

Ernesto Alonso Gutiérrez

Physical data acquisition and PWM signal generation using a microcontroller allowing control for DC motor rotation speed

Helsinki Metropolia University of Applied Sciences
Bachelor's degree
Electronics
Bachelor's Thesis
May 2012

PROYECTO FIN DE CARRERA

TEMA: Sistema de control

TÍTULO: Adquisición de datos físicos y generación de señal PWM mediante microcontrolador para controlar velocidad de rotación de motor DC

AUTOR: Ernesto Alonso Gutiérrez

TUTOR: Thierry Baills

DEPARTAMENTO: Electronics

CENTRO DE LECTURA: Metropolia Ammattikorkeakoulu

Fecha de Lectura: 6 Junio, 2012

Calificación:

RESUMEN DEL PROYECTO:

El objetivo de este proyecto es diseñar un sistema capaz de controlar la velocidad de rotación de un motor DC en función del valor de temperatura obtenido de un sensor. Para ello se generará con un microcontrolador una señal PWM, cuyo ciclo de trabajo estará en función de la temperatura medida.

En lo que respecta a la fase de diseño, hay dos partes claramente diferenciadas, relativas al hardware y al software.

En cuanto al diseño del hardware puede hacerse a su vez una división en dos partes.

En primer lugar, hubo que diseñar la circuitería necesaria para adaptar los niveles de tensión entregados por el sensor de temperatura a los niveles requeridos por ADC, requerido para digitalizar la información para su posterior procesamiento por parte del microcontrolador. Por tanto hubo que diseñar capaz de corregir el offset y la pendiente de la función tensión-temperatura del sensor, a fin de adaptarlo al rango de tensión requerido por el ADC.

Por otro lado, hubo que diseñar el circuito encargado de controlar la velocidad de rotación del motor. Este circuito estará basado en un transistor MOSFET en conmutación, controlado mediante una señal PWM como se mencionó anteriormente. De esta manera, al variar el ciclo de trabajo de la señal PWM, variará de manera proporcional la tensión que cae en el motor, y por tanto su velocidad de rotación.

En cuanto al diseño del software, se programó el microcontrolador para que generase una señal PWM en uno de sus pines en función del valor entregado por el ADC, a cuya entrada está conectada la tensión obtenida del circuito creado para adaptar la tensión generada por el sensor. Así mismo, se utiliza el microcontrolador para representar el valor de temperatura obtenido en una pantalla LCD. Para este proyecto se eligió una placa de desarrollo mbed, que incluye el microcontrolador integrado, debido a que facilita la tarea del prototipado.

Posteriormente se procedió a la integración de ambas partes, y testeado del sistema para comprobar su correcto funcionamiento. Puesto que el resultado depende de la temperatura medida, fue necesario simular variaciones en ésta, para así comprobar los resultados obtenidos a distintas temperaturas. Para este propósito se empleó una bomba de aire caliente.

Una vez comprobado el funcionamiento, como último paso se diseñó la placa de circuito impreso.

Como conclusión, se consiguió desarrollar un sistema con un nivel de exactitud y precisión aceptable, en base a las limitaciones del sistema.

Author	Ernesto Alonso Gutiérrez
Title	Physical data acquisition and PWM signal generation using a microcontroller allowing control for DC motor rotation speed
Number of pages	41 + 6 appendices
Date	June 6 th , 2012
Degree	Bachelor of Engineering
Degree Programme	Electronics
Instructor	Thierry Bails
<p>It is obvious that day by day people's daily life depends more on technology and science. Tasks tend to be done automatically, making them simpler and as a result, user life is more comfortable. Every single task that can be controlled has an electronic system behind.</p> <p>In this project, a control system based on a microcontroller was designed for a fan, allowing it to go faster when temperature rises or slowing down as the environment gets colder. For this purpose, a microcontroller was programmed to generate a signal, to control the rotation speed of the fan depending on the data acquired from a temperature sensor.</p> <p>After testing the whole design developed in the laboratory, the next step taken was to build a prototype, which allows future improvements in the system that are discussed in the corresponding section of the thesis.</p>	
Keywords	Temperature sensor, data acquisition, PWM, DC motor, microcontroller

Preface

First of all, I want to thank Metropolia University, in particular the head of the electronics program, Heikki Valmu, for letting me do this project and to Thierry Bails for supervising my work to keep it on the right track.

My deepest thanks to my family, specially my parents, for supporting me on the decision of going abroad, encouraging me when I was full of doubts.

Thanks also to my friends, for being by my side somehow although there were thousands of kilometers between us.

Also a little piece of this book belongs to my dear “telecos”, for making the adventure of studying a degree during the years I have spent at university something unforgettable full of good moments.

And last but not least, to my Erasmus friends, for sharing with me this incredible experience and make me enjoy every single moment I have been in Finland.

Helsinki. June 06th, 2012

Ernesto Alonso Gutiérrez

Content

List of Figures

List of Abbreviations

1	Introduction	1
2	Theoretical Background	3
2.1	System Requirements	3
3	System Development	8
3.1	Hardware Development	8
3.1.1	<i>Temperature Sensor</i>	8
3.1.2	DC Motor	16
3.2	Software Development	18
3.2.1	<i>Microcontroller</i>	18
3.2.2	<i>Compiler Tools</i>	19
3.2.3	<i>Code Writing</i>	21
4	Implementation and Testing	23
4.1	Temperature Sensor	24
4.2	DC Motor	26
4.3	LCD Display	28
5	Board Designing	30
5.1	Schematic	30
5.2	Layout	31
5.3	CAM and Printing	33
6	Materials	35
7	Discussion	37
7.1	Results and Issues	37
7.2	Future Development	38
8	Conclusion	39
	References	40

Appendices

Appendix 1. Temperature Sensor LM335 Datasheet

Appendix 2. ADC Characteristics

Appendix 3. Temperature sensor circuitry

Appendix 4. Source Code

Appendix 5. Mbed board schematic

Appendix 6. DC Motor (fan) datasheet

List of Figures

Figure 1. Block diagram of the project	4
Figure 2. Analog to Digital conversion. Reprinted from Len Staller [5]	5
Figure 3. PWM signals, with different duty cycles. Reprinted from National Instrument	6
Figure 4. Block diagram of the project	7
Figure 5. LM335 temperature sensor calibration circuit. Reprinted from National Semiconductor	8
Figure 6. Current source for the temperature sensor.....	9
Figure 7. LM335 voltage range vs temperature.....	10
Figure 8. Temperature sensor function, with and without offset.....	11
Figure 9. Circuit to adapt the temperature sensor voltage to ADC input.....	12
Figure 10. Voltage regulator circuit.....	15
Figure 11. DC motor control circuit.....	17
Figure 12. Mbed board. Reprinted from mbed	19
Figure 13. Screenshot taken from the mbed compiler	20
Figure 14. Screenshot taken from the mbed compiler	20
Figure 15. Screenshot taken from Keil uVision 4	21
Figure 16. Breadboard with the system implemented on it	23
Figure 17. Block diagram with the connection of the tools used for testing	25
Figure 18. LM335 temperature sensor response in time	26
Figure 19. Oscilloscope screenshots.....	27
Figure 20. Diagram of connecting points of the oscilloscope probes	28
Figure 21. LCD Display testing	29
Figure 22. Screenshot from PADS Logic showing the schematic file.....	31

Figure 23. Screenshot from PADS Layout showing the components once rearranged on the board.....	32
Figure 24. Screenshot from PADS Layout showing the board once routed	32
Figure 25. Screenshot from PADS Layout showing the board after creating the ground plane.....	33
Figure 26. Screenshot from PADS Layout. Preview of how the design will look on the board viewed from top layer.....	34
Figure 27. Hot-air station Hakko 850. Reprinted from Hakko USA	35
Figure 28. Multimeter with thermocouple connected(left). Detail of the thermocouple connector (right)	36
Figure 29. System working at room temperature.....	39

List of Abbreviations

ADC	Analogue-Digital Converter
DC	Direct Current
LCD	Liquid Crystal Display
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
PCB	Printed Circuit Board
PWM	Pulse Width Modulator
R	Resistor
RPM	Revolutions per minute
RTD	Resistance temperature detector
SDK	Software Development Kit

1 Introduction

Technology has allowed human beings to simplify and automate tasks, making their life more comfortable and easier. We are surrounded by control systems in daily life. It is obvious that the way of life has experienced more changes in the last decades than in several centuries before. Now, it is possible to perform several tasks just by pressing a button, leaving for technology the part of controlling the process and the effort of doing it.

This trend has increased significantly with the appearance, and its later development, of integrated circuits in the second half of 20th century, which meant that it was possible to make more complex electronic devices with an affordable price. There was also a noticeable change with the appearance of digital devices, especially with the first microprocessor. [1.]

Using a microcontroller, a microprocessor with several electronic peripherals all embedded into the same chip, permits to perform several tasks at the same time, with a higher working frequency, which means that the purpose for which the system has been designed is reached in a simpler and faster way.

By the mid-seventies people began to realize that you could get a microprocessor for very little money and do some very phenomenal things with it.

Robert Noyce. Intel [1, 110]

The goal of this project is to develop a system, able to control the rotation speed of a DC motor, and by such way demonstrate how a system with microcontroller architecture can allow creating an accurate control system.

In particular, the objective of the project is to generate a PWM signal with a duty cycle as a function of the temperature value obtained from a sensor, in order to control the speed rotation of a DC motor that moves a fan. This way it goes faster as the temperature rises, in order to cool down the environment where the system is implemented. As a result, the user does not have to check constantly whether it is getting hot or not, as the system itself controls the temperature.

This report explains the different steps followed during the designing, developing and testing of the system.

2 Theoretical Background

The starting point in this project is that analogue systems are limited as they cannot carry a huge amount of data as it is required nowadays to get more accurate results. Here is where digital signals appear, because when information is converted into binary data instead of electrical pulses, is possible to process that information precisely.

That is the reason for using a microcontroller, they allow to process data in an easier and faster way, in order to obtain accuracy in measurements, unreachable in analogue systems. But then comes the problem, analogue signals have to be converted into digital ones.

2.1 System Requirements

In the first approach to the project, the main idea is to change the rotation speed of a DC motor that moves a fan, according to temperature. In another words, depending of the information got from a temperature sensor, a signal should be created to control the rotation speed of the motor.

As mentioned before, to get more accurate control over the system, it is preferable to use digital technology. So, the main task to be done is converting the analogue signal obtained from the sensor, carrying the information of the temperature, and converting it into a digital one, so that a microcontroller can analyze it and create another signal that will control the motor.

As temperature is measured, via a sensor, the signal obtained is analogue, but microcontrollers work with binary numbers, so the signal provided by the sensor requires to be converted into a digital one. For that purpose an analogue-to-digital converter (ADC) is needed.

With the DC motor, information has to change the other way round, is required to convert the binary data into an electric pulse in order to move the motor. The rotation speed of a DC motor is proportional to the voltage supplied. If motor were ideal, when applying a voltage, it would draw infinite current. But it has certain limitations during the start-up, so applying a voltage would cause excessive torque, which can break the motor, and excessive current which might burn the circuitry. The solution chosen for this is to use a MOSFET transistor, switching at a certain frequency, limiting the voltage that is supplied to the motor. To make the MOSFET transistor work as a switch, a PWM signal should be applied in the gate terminal of the transistor, so it changes from saturation to cut off [2].

Taking all that into consideration, the first block diagram of the project is the one shown in figure 1.

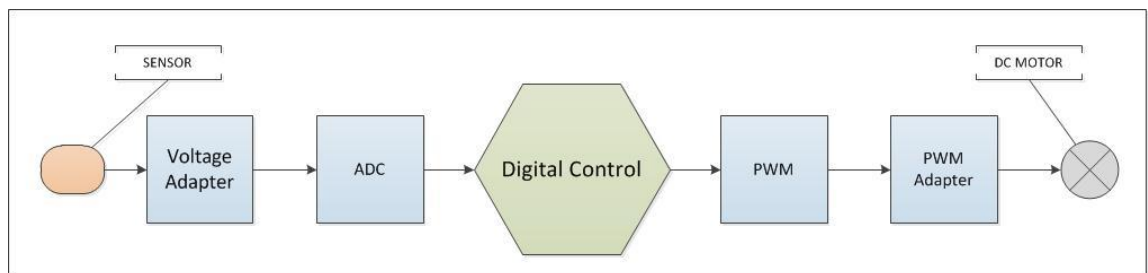


Figure 1. Block diagram of the project

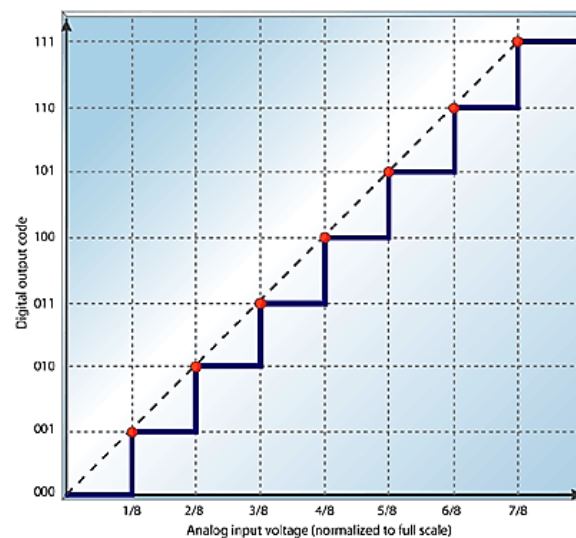
Taking a deeper look to the different blocks of the diagram, the first of them is the temperature sensor. There are different types, being RTD, thermocouples, thermistors and silicon based sensors the most used ones. Any of these types is not appropriate for all applications, as they have different characteristics, as can be seen in table 1.

The decision of using a silicon based sensor was based on the requirements of the project, which will be further explained in the next section.

Table 1.Types of temperature sensor and their characteristics.**Reprinted from Bonnie Baker [3]**

TABLE 1 POPULAR TEMPERATURE SENSORS				
Parameter	Thermocouple	RTD	Thermistor	Silicon based
Temperature range (°C)	–270 to +1800	–250 to +900	–100 to +450	–55 to +150
Sensitivity	Tens of microvolts per degree Celsius	0.00385Ω/Ω/°C (platinum)	Several Ω/Ω/°C	Uses technology that is approximately –2 mV/°C sensitive
Accuracy (°C)	±0.5	±0.01	±0.1	±1
Linearity	Requires at least a fourth-order polynomial or equivalent look-up table	Requires at least a second-order polynomial or equivalent look-up table	Requires at least a third-order polynomial or equivalent look-up table	At best within ±1°C; no linearization required unless higher accuracy is desired; high accuracy results require a third-order polynomial
Ruggedness	Rugged due to larger-gauge wires and insulation materials, which enhance sturdiness	Susceptible to damage as a result of vibration due to #26 to #30 AWG leads, which are prone to breakage	Thermistor element is housed in a variety of ways; however, the most stable hermetic units are enclosed in glass; generally, thermistors are difficult to handle, but shock and vibration do not affect them	As rugged as any other IC in a plastic package, such as dual-in-line or surface-outline ICs
Responsiveness in stirred oil (sec)	Less than 1	1 to 10	1 to 5	4 to 60
Excitation	None required	Current source	Voltage source	Typically, supply voltage
Form of output	Voltage	Resistance	Resistance	Voltage, current, or digital
Typical size	Bead diameter is five times wire diameter	0.25×0.25 in.	0.1×0.1 in.	From TO-18 transistors to plastic DIPs
Price	\$1 to \$50	\$25 to \$1000	\$2 to \$10	\$1 to \$10

The ADC converter takes samples of the analogue signal, and then by comparing it to a reference voltage, a binary digit is obtained, in another way, a digital signal [4]. The ADC transfer characteristic is shown graphically in figure 2.

**Figure 2. Analog to Digital conversion. Reprinted from Len Staller [5]**

Nowadays it is possible to find ADCs already embedded into another component, working as a peripheral of it. This is the case of almost all the microcontrollers that are for sale right now.

Another feature microcontrollers provide is that they can generate a PWM signal, which is a square pulse wave. They are really useful to control analogue systems, in this case a MOSFET based one, because even though they are still a digital signal, they act as a kind of switch, as they change from ground to supply voltage cyclically, varying the amount of energy the load receives. This amount can be controlled by changing the duty cycle that is the time the pulse is high.

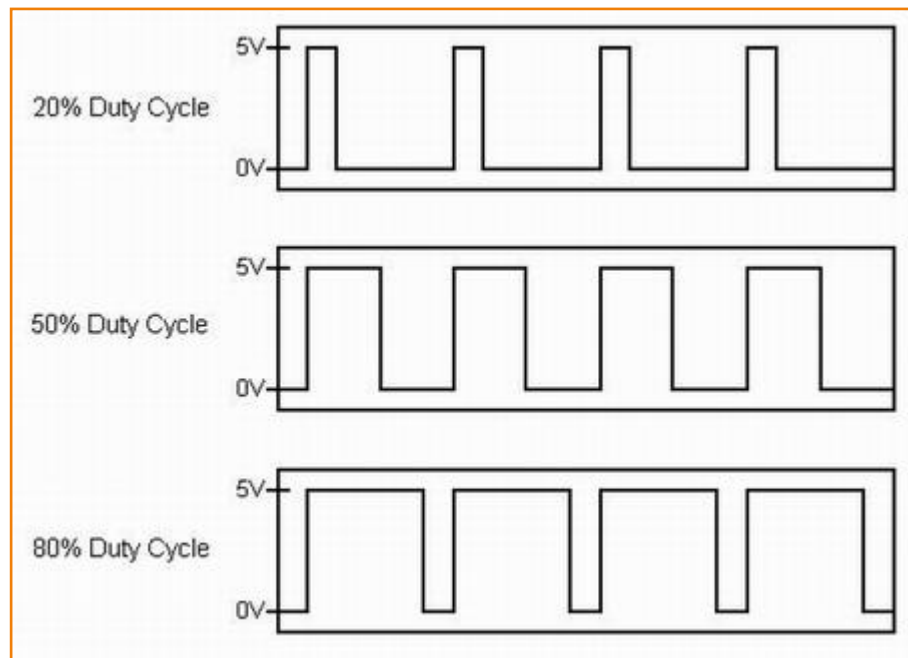


Figure 3. PWM signals, with different duty cycles. Reprinted from National Instruments [6]

As a result, by changing the duty cycle of the PWM signal, the voltage supplied to the motor will also change, controlling this way the rotation speed of the motor.

What microcontrollers can do is generating this kind of signal according to digital data given, in this case, the binary values given by the ADC. As a result, the PWM signal obtained will change its duty cycle in function of the temperature. If this signal controls the DC motor, then it will have a rotation speed that depends on temperature.

So, the problem is narrowed. Now it consists of adapting the voltage given by the sensor to the voltage requirements of the ADC, information that can be found into the microcontroller's datasheet, and then using the digital code obtained to generate a PWM signal with the microcontroller.

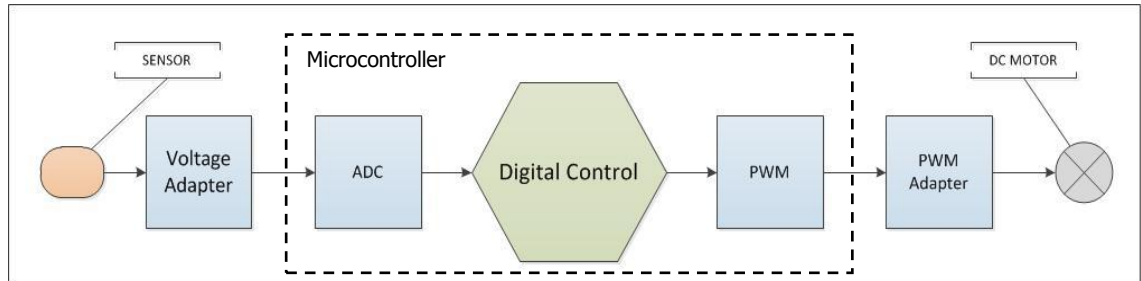


Figure 4. Block diagram of the project

The solution given will be studied in depth in the next chapter of the thesis, but the first approach is shown as a block diagram in figure 4.

3 System Development

3.1 Hardware Development

This part of the project was divided into two smaller ones, clearly distinguishable. The first one focuses on the temperature sensor and the electronics necessary to adapt its voltage to the ADC input. The second one relates to the DC motor, and it consists of converting the PWM signal, obtained as a function of the temperature, into a pulse in order to control the rotation speed of the motor mentioned before.

3.1.1 Temperature Sensor

The temperature sensor chosen for this project is LM335, from Texas Instruments. It was chosen among others because of its accuracy, easy calibration, and its low cost. Basically it can be considered as a zener diode with an adjustment pin, which means it is an electronic component that does not require complex electronics surrounding it or calculations in order to work. It is only required to have a fix current value going through it, to obtain a voltage as function of temperature in the output pin. In figure 5 an extract of the information given by the manufacturer in the datasheet can be seen.

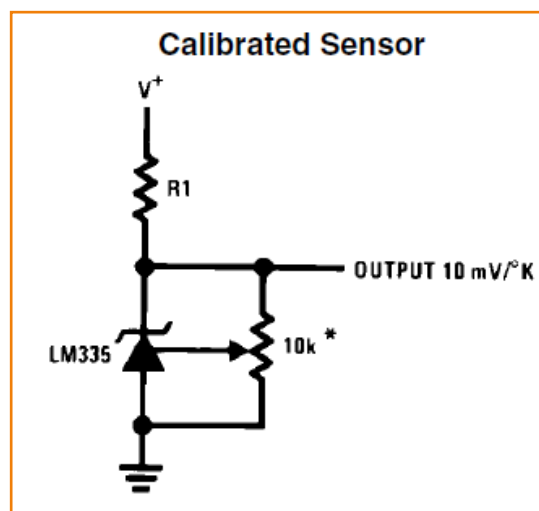


Figure 5. LM335 temperature sensor calibration circuit. Reprinted from National Semiconductor [7]

For that purpose, a current source was designed, shown below in figure 6. Having a fix voltage (V_{ctrl}), the current through the sensor is the same as in the resistor. This option was chosen because the current value going through the sensor is fixed, it does not depend on the voltage value on the sensor, as it happens in the circuit shown above in figure 5. As can be seen in the picture below, the value of the current will only depend on the values of control voltage and the resistor.

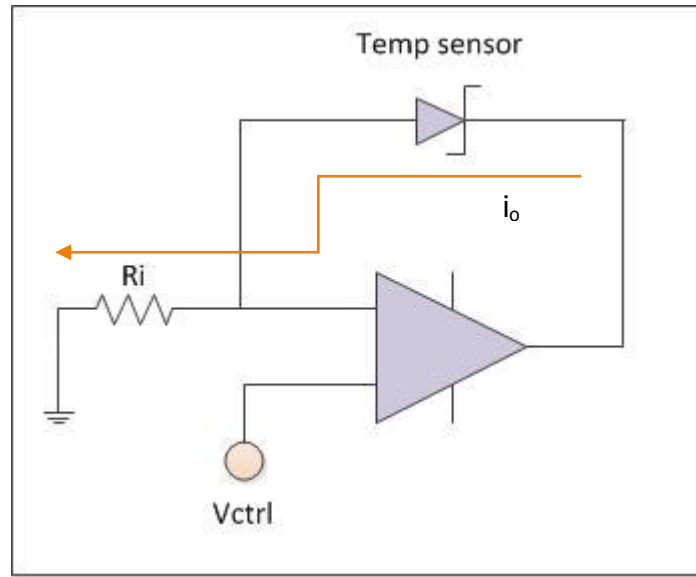


Figure 6. Current source for the temperature sensor

The equation to calculate the value of the current is:

$$i_o = \frac{v_{ctrl}}{R_i}$$

From the LM335 datasheet, included in appendix 1, it is possible to make a graph to determine the relationship between temperature and the voltage given by the sensor, as can be shown in figure 4. According to the data given by the manufacturer, the sensor will give a voltage of 2.98 V at 25° C, and it increases by 10 mV with every degree the temperature rises. The system is required to work in a range from 0° C to 100° C. Taking this into consideration, the equation for the graph shown below is:

$$V_o = 2.98 \text{ V} + (10 \cdot 10^{-3} \text{ V} \cdot (T - 25^\circ \text{ C})) \quad (1)$$

Transferring this into the graph, it can be seen in figure 7 that the voltage expected at the output of the sensor, once calibrated, has to be in a range from 2.73 V to 3.73 V.

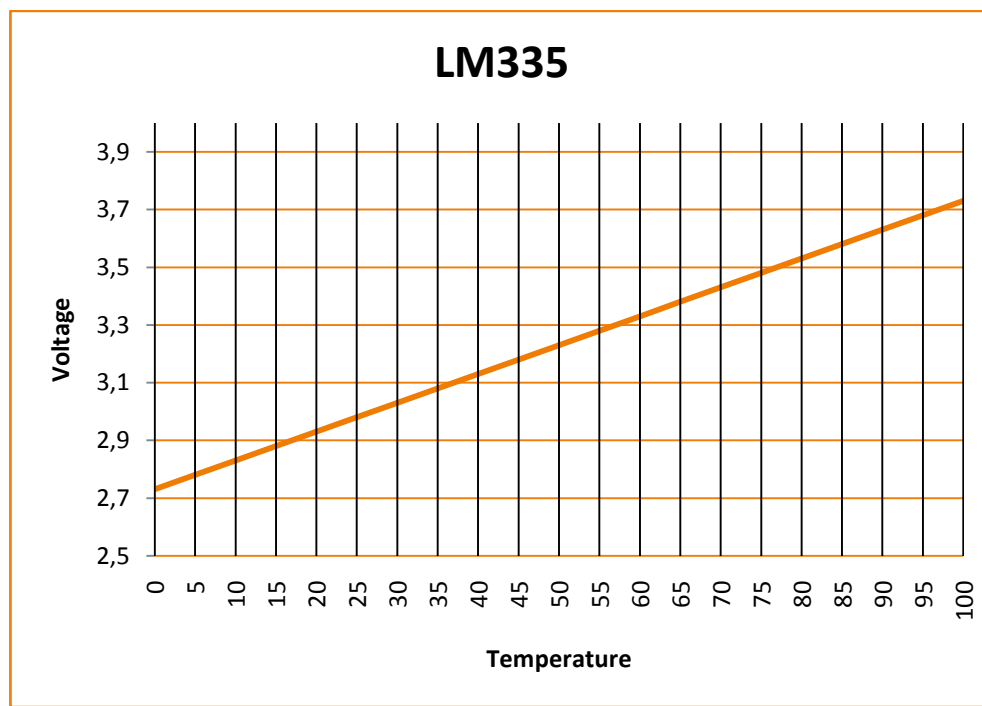


Figure 7. LM335 voltage range vs temperature

As it was explained in the previous section, the next step is to give this voltage to the ADC, to obtain a binary number as function of the temperature.

The microcontroller datasheet shows that the maximum voltage that can be applied to the ADC input is 3.3 V, information that can be checked in appendix 2. It would be also interesting to use the whole ADC operating range, which means the signal given goes from 0 to 3.3 V, obtaining a more accurate temperature measurement.

But as explained recently, the temperature sensor chosen for this project can reach a maximum voltage value of 3.73 V, which is higher than 3.3 V. So, the idea is to design a circuit that gives an output voltage from 0 to 3.3V when applying an input voltage that goes from 2.73 to 3.73 V given by the sensor.

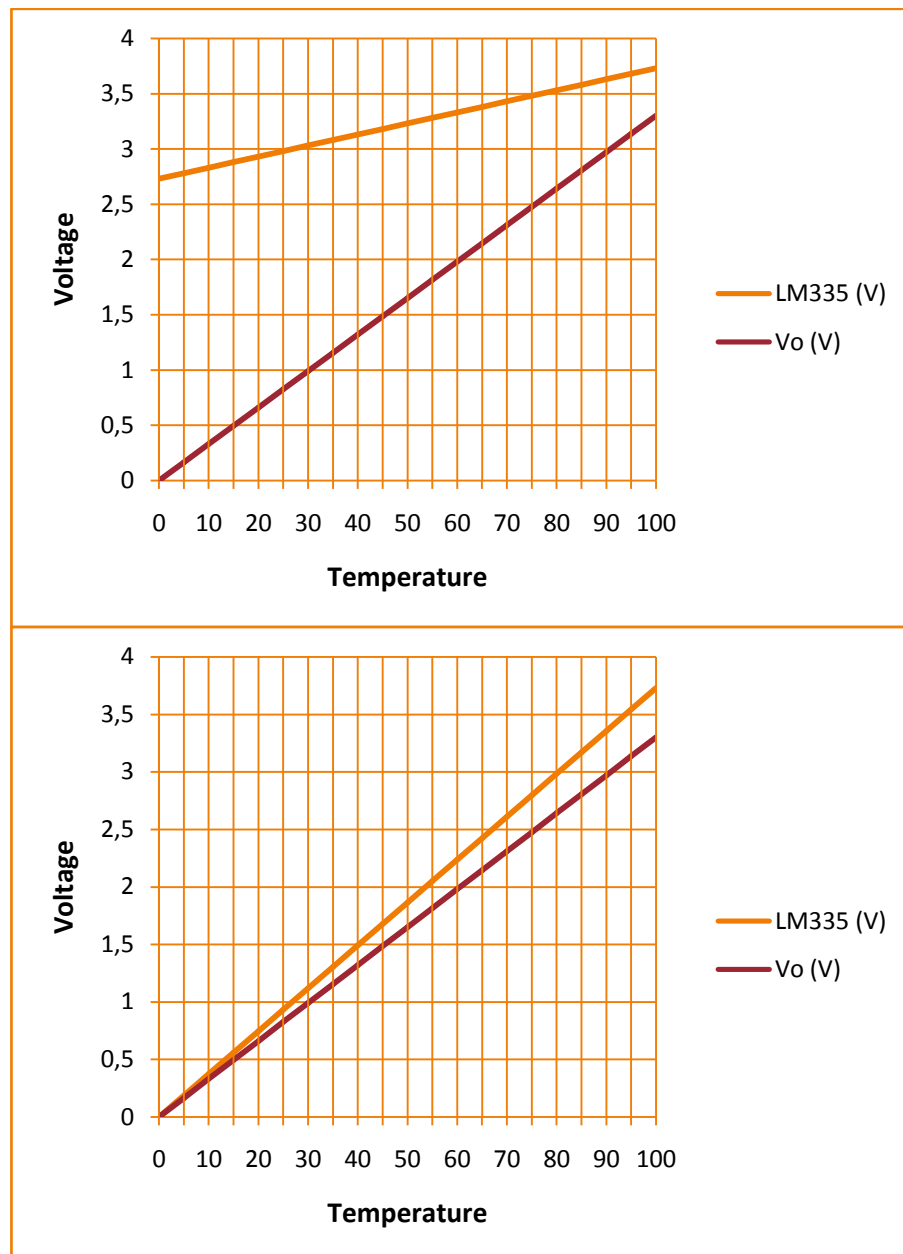


Figure 8. Temperature sensor function, with and without offset

For that purpose, a circuit was designed, allowing the adjusting of the offset value, as can be seen in the figure above, and then the slope of the signal obtained from the temperature sensor.

After taking several solutions into consideration, the one adopted is shown in the upcoming figure.

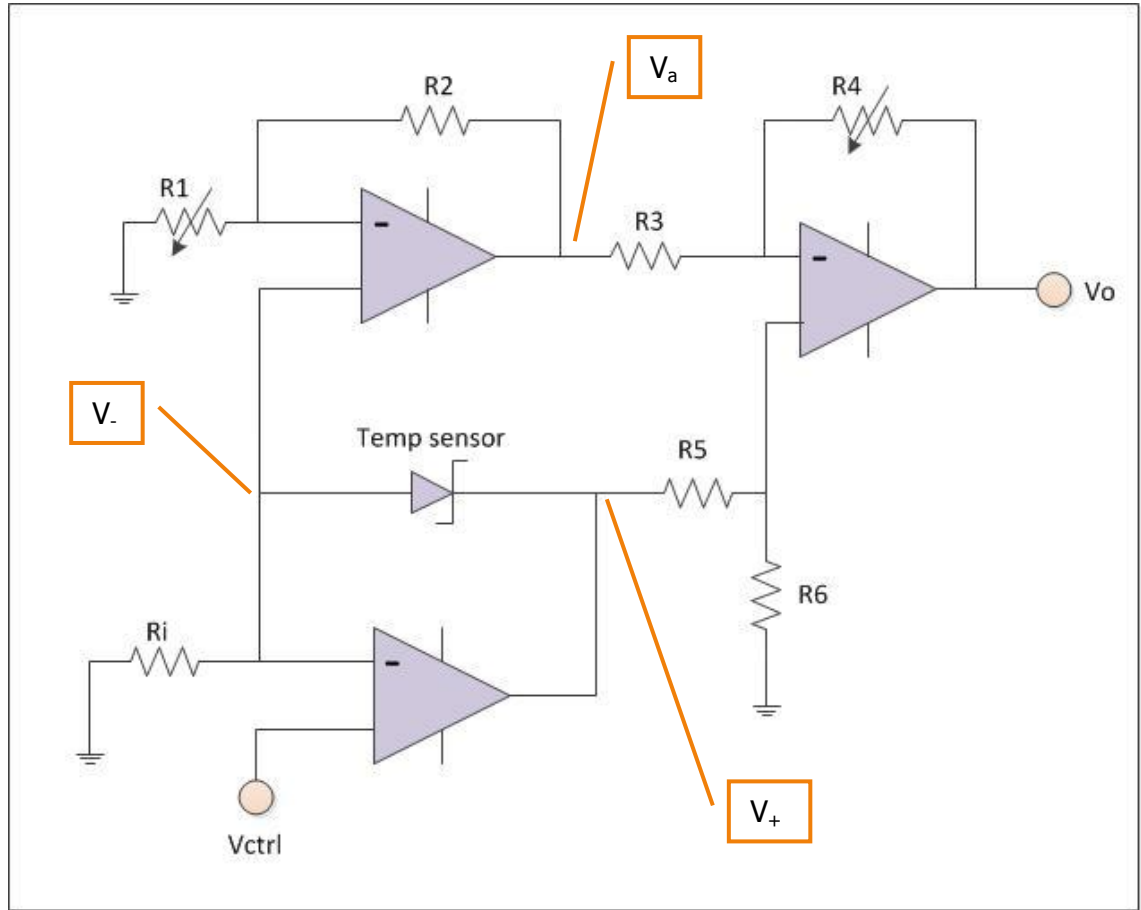


Figure 9. Circuit to adapt the temperature sensor voltage to ADC input

Analyzing the circuit shown above, is possible to obtain the mathematical expression of the output voltage:

$$V_- = V_a \frac{R_2}{R_1 + R_2} \rightarrow V_a = V_- \left(1 + \frac{R_1}{R_2} \right) \quad (2)$$

Then, if $R_5/R_6 = R_3/R_4$

$$V_o = V_+ \frac{R_4}{R_5} - V_a \frac{R_4}{R_5} \quad (3)$$

$$V_o = V_+ \frac{R_4}{R_5} - V_- \left(1 + \frac{R_2}{R_1} \right) \frac{R_4}{R_5} \quad (4)$$

$$V_o = V_+ \frac{R_4}{R_5} - V_- \left(\frac{R_4}{R_5} + \frac{R_2 R_4}{R_1 R_5} \right) \quad (5)$$

If $R_2 = R_5$ then the equation is simplified. In this project $1k\Omega$ resistors were chosen. Also, as can be shown in figure 9, the voltage in V_- is the same used as V_{ctrl} being then the voltage in V_+ the voltage mentioned before plus the voltage given by the sensor. Then:

$$V_o = (V_{CTRL} + V_{SENSOR}) \frac{R_4}{1k} - V_{CTRL} \left(\frac{R_4}{1k} + \frac{R_4}{R_1} \right) \quad (6)$$

@ 0°C

$$0 = (V_{CTRL} + 2.73) \frac{R_4}{1k} - V_{CTRL} \left(\frac{R_4}{1k} + \frac{R_4}{R_1} \right) \quad (7)$$

$$V_{CTRL} = 0.00273 R_1 \quad (8)$$

Offset can be set by changing the value of resistor R_1

@ 100°C

$$3,3 = (V_{CTRL} + 3.73) \frac{R_4}{1k} - V_{CTRL} \left(\frac{R_4}{1k} + \frac{R_4}{R_1} \right) \quad (9)$$

$$0 = V_{CTRL} - 0.00373 R_1 + 3,3 \frac{R_1}{R_4} \quad (10)$$

Having fixed the value of resistor R_1 in the first step, to fix the slope it is only necessary to change the value of resistor R_4 . Expressed as a general function:

$$0 = V_{CTRL} - \alpha R_1 + \beta \frac{R_1}{R_4} \quad (11)$$

Where α depends on the sensor, and β depends on the ADC maximum range.

As can be noticed, the values obtained with the equation before also depend on the value of the voltage chosen as control. In order to pick a proper value, it is necessary to go back to the schematic shown in figure 9.

The maximum output voltage that an operational amplifier can give is more or less the same used as power source, in this case, +5 V. As it was mentioned before, the output voltage V_+ is the voltage used as control plus the voltage given by the sensor. Figure 7 tells that the maximum voltage the temperature sensor LM335 will give is 3.73 V. Then:

$$5 - 3.73 = 1.27 \text{ V} \quad (12)$$

That is the maximum value the control voltage can have.

Going back to equations (8) and (11), it can be seen that the values of the control voltage (V_{CTRL}), α and β , are known, so it is possible to calculate now the values of the resistors R_1 and R_4 .

Table2.Values obtained for the resistors according to the parameters

SETTINGS							
	T_a	V_{CTRL}	ADC Max	α	β (ADC Out)	R1 (offset)	R4 (slope)
1st STEP	0	1,27	3,3	0,00273	-	465,20	-
2nd STEP	100	1,27	3,3	0,00373	3,3	-	3300,00

In order to obtain the control voltage, it was required to design a voltage regulator. This kind of circuit generates a constant voltage in its output. In certain electronic circuits this is very important because just a small change may vary the results obtained significantly.

This option was taken, because even though it is possible to find regulators with low voltage output, typical ones have higher output voltages. As an example, at the laboratory the smallest one available was 3.3 V, roughly twice the value needed. The downside is that the voltage created will not be as accurate as with a commercial regulator, due to the value and tolerance of the components chosen.

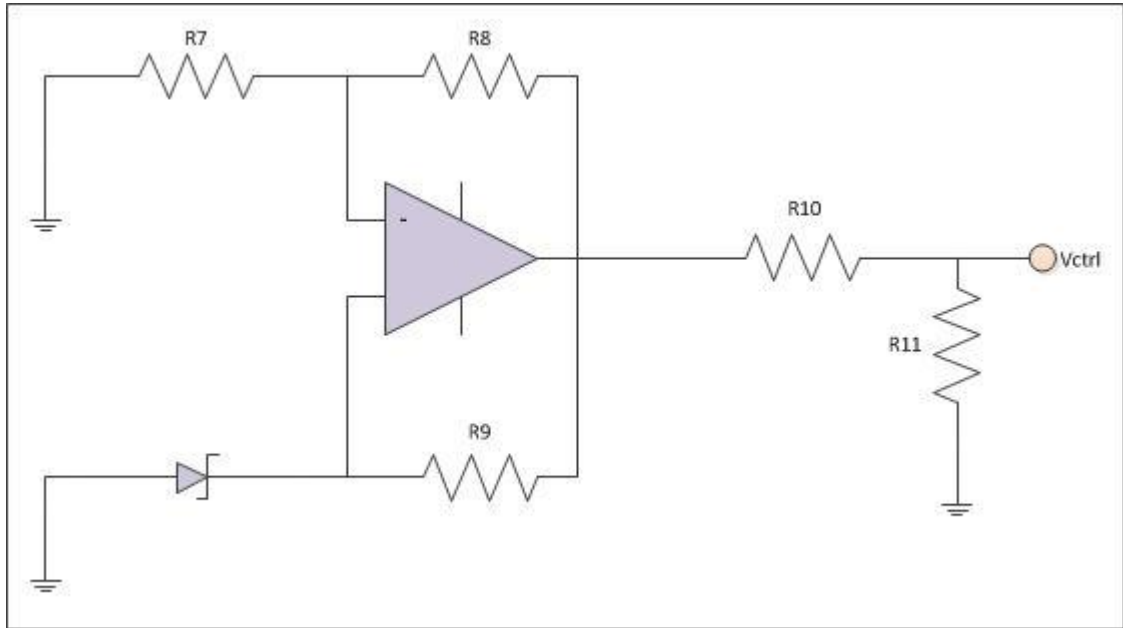


Figure 10. Voltage regulator circuit

The circuit designed for this purpose can be seen in figure 10. The calculations to choose the values of the components are shown below.

$$3.3 = 5 \cdot \frac{R_7}{R_7 + R_8} \rightarrow \frac{R_7}{R_8} = 2 \quad (13)$$

$$1.27 \cong 1.25 = 5 \cdot \frac{R_{11}}{R_{10} + R_{11}} \rightarrow \frac{R_{10}}{R_{11}} = 3 \quad (14)$$

All the results obtained with the calculations made in this part are summarized in the table included in appendix 2.

3.1.2 DC Motor

The fan used in this project is moved due to a low power DC motor. This motor needs a power supply to be moved. The signal provided by the microcontroller is a PWM one, so to transform this type of signal into a voltage to control the motor rotation speed the solution taken consists of using a MOSFET transistor, as it was explained in the previous part.

Depending on the frequency and the duty cycle of the PWM signal, the MOSFET transistor switches from saturation region to cut off fast enough, generating a current passing through the transistor from drain to source. This current will control the torque of the motor, therefore the current the motor will draw. [8]

$$Torque = K_T \cdot i_{ds} \quad (15)$$

When MOSFET is active, a voltage is created from drain to source (V_{ds}), making it possible to control the voltage that drops in the DC motor, controlling its speed. [8]

$$RPM = K_V \cdot V_{motor} \quad (16)$$

Both, torque and RPM, are determined from the motor constants K_T (torque constant) and K_V (voltage constant).

As explained in the theoretical background section, the duty cycle will vary depending on the temperature value measured, so it will go from 0 to 100% as the temperature range also goes from 0 to 100° C.

About the frequency of the signal, a value of 100 kHz was chosen empirically. This value was chosen because it should be high, as it is being used to control a DC motor, so the changes in the signal must be fast enough to make the current look continuous, and not a series of pulses. Another important reason to choose a high frequency value is that if it is above 20 kHz, the noise generated by the motor is out of the human hearing range, so they cannot be noticed [2].

Another advantage obtained using this method is that power consumption is reduced significantly in the MOSFET transistor, which means there is not a huge amount of power to be dissipated, and as a result, no temperature rising that may damage the transistor due to a thermal overload [2].

There are also drawbacks in using a high frequency, and one is that switching energy losses in the MOSFET are higher. To reduce these losses, a capacitor was connected in the output of the MOSFET, as well as a protection diode.

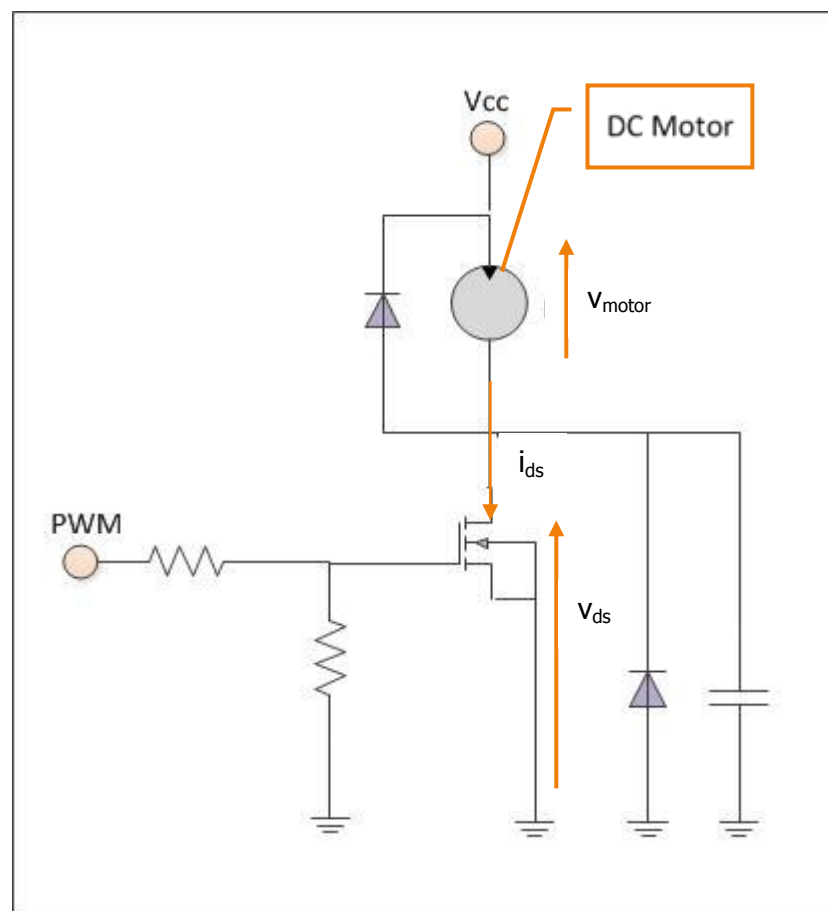


Figure 11. DC motor control circuit

Also, as a protection, a flywheel diode was connected across the inductive load of the DC motor, to dissipate possible electromotive forces generated when the MOSFET transistor turns off [2]. This diode can be seen in figure 11, connected in parallel with the DC motor.

As the motor is running forward all time, it does not need to reverse rotation, only one MOSFET is required to control the motor. This simplifies a lot the solution, as the complexity of the circuit needed, in case it was necessary to reverse motor rotation, would have risen noticeably. The circuit designed is shown in figure 11.

3.2 Software Development

In this part of the project, the aim is to obtain a PWM signal as a function of the binary data given by the ADC. Then, this data containing the information about the temperature measured is also shown in an LCD display, in Celsius and Fahrenheit degrees. All this tasks are performed by a microcontroller, which needs to be programmed requiring a code, in C language in this case, to be written.

3.2.1 Microcontroller

For this purpose it was needed to choose a microcontroller that suits the requirements of the present project. This means a microcontroller including ADC and PWM generator among its peripherals. Nowadays almost all devices of this kind meet this requirement, although there are some which do not, as for example some PIC microcontrollers.

It was decided to use a board with a microcontroller already embedded on it, called mbed, manufactured by ARM that is shown in figure 12. As it is said in the mbed official webpage:

"The mbed Microcontrollers are a series of microcontrollers development boards designed for fast, flexible and low-risk and professional rapid prototyping."



Figure 12. Mbed board. Reprinted from mbed [9]

It is obvious that for the purpose pursued in this project, this tool is very appropriate. When developing a prototype the programmer needs a fast, easy-changeable hardware, because a lot of changes are required. Another feature that influenced the decision of choosing this tool was that the SDK environment includes some libraries that simplify the task of programming because it is possible to forget about low-level code and datasheets, focusing on the code that will provide the solution itself. [10]

3.2.2 Compiler Tools

As previously said, a code needs to be written to program the microcontroller in order to perform the operations that will create the PWM signal from the data given by the ADC. To perform this, an interface is necessary to write and compile the code mentioned.

This is another of the reasons why the mbed board was chosen. The developer offers an online compiler, which can be seen in figure 13, which allows the programmer to write and test the code in a quick way. No installations are required, as it is a web application, so it is possible to access it from any computer with an internet connection [11].

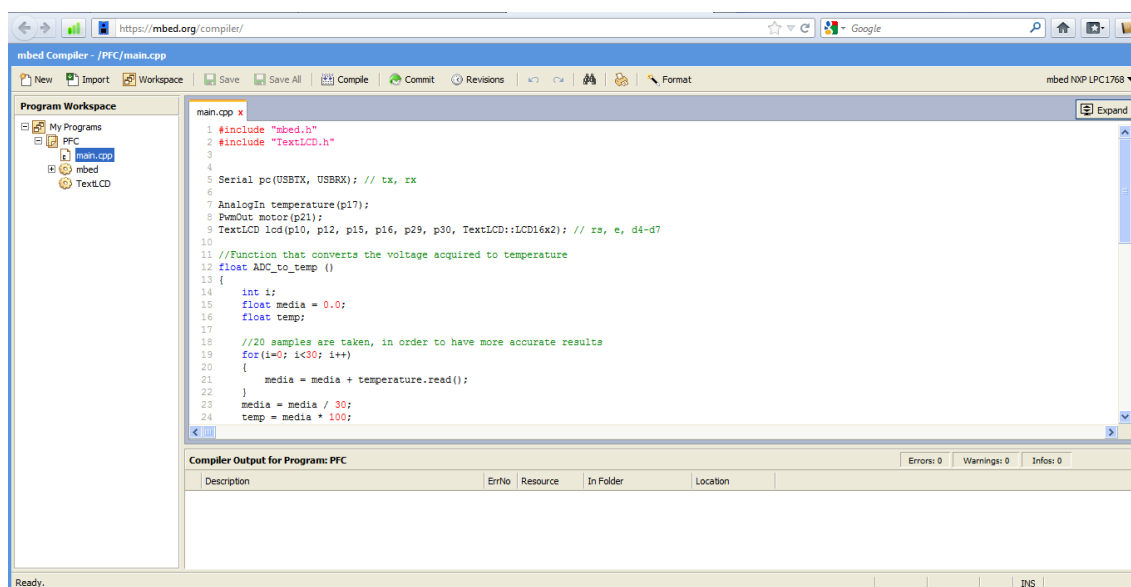


Figure 13. Screenshot taken from the mbed compiler [11]

This tool is very appropriate for quick development, but has a lack of features that might be useful in case a deeper debugging is required, as well as is interesting to have the possibility of using an offline compiler, in case no internet connection is available.

For this reason, mbed offers the possibility of exporting the project to another program, and so was done in several times. In figure 14, this characteristic is shown.

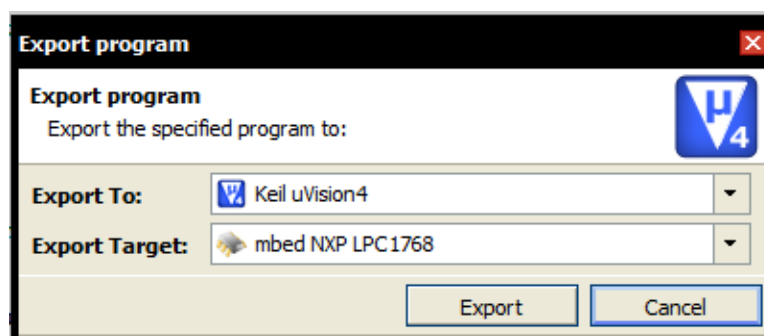


Figure 14. Screenshot taken from the mbed compiler [11]

It obviously helps a lot to have to different tools, which complement each other, available to develop the code. This second tool mentioned is Keil uVision 4, which offers a large amount of characteristics for debugging, as well as no internet connection is required to use it. Can be seen in figure 15.

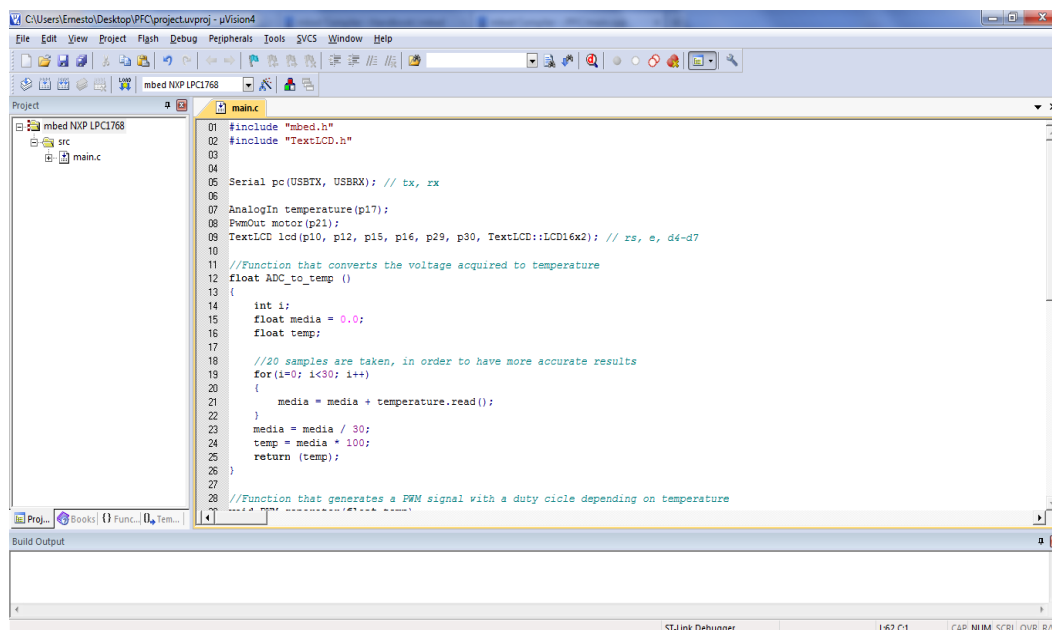


Figure 15. Screenshot taken from Keil uVision 4 [12]

3.2.3 Code Writing

As mentioned before, another reason the mbed board was chosen for this project is that there are some libraries, high-level code interfaces to control the peripherals, available online, allowing users to save time looking datasheets, focusing on developing the system. [13]

Due to the reason that libraries provided by mbed are coded in C++ instead of C, it was completely necessary to use this programming language that being quite similar to C, did not suppose a challenge.

The voltage generated in the previous stage, adapting the one given by the temperature sensor into another one that suits better the ADC, is connected to one of the analog inputs included into the mbed board that contains the microcontroller. Then, thanks to one of the libraries, it is possible to obtain the digital data just calling a function. In order to avoid possible mistakes when measuring, several samples are taken and then the average value of all of them is the one taken into consideration in the rest of the code.

Then, as the range of temperatures measured by the sensor goes from 0 to 100° C, it is very easy to obtain a PWM signal where the duty cycle would coincide with the temperature measured before. Again, using one of the libraries included, it was only necessary to call a function to complete this task. The frequency of the PWM signal is also set in the code, using another function provided on the libraries.

In the last step, the value of the temperature obtained before, which is in Celsius degrees, is converted into Fahrenheit ones. Then both values are shown into the LCD display. Once again, the task was easily solved thanks to another library provided by the mbed SDK environment.

As the system is running continuously at a frequency unnoticeable for human eyes, it was necessary to include a delay, so data shown in the display is refreshed every 5 seconds. It was considered enough, as temperature is a parameter that in real situations does not have abrupt changes.

The whole code can be seen in appendix 3. As well, the schematic file of the mbed board, needed to choose the pins for the different signals, is included in appendix 4.

4 Implementation and Testing

One of the most important facts to take into account, when designing electronics is that sometimes real values may differ from theoretical ones. This happens because some characteristics of the components are not taken into consideration, which is called modelization, while calculating the variables involved in the design, there is always a difference between ideal values and real ones.

After developing the design, the next step to be taken is to prove that all the theoretical background behind it, when is taken into reality, really works as expected or if it is necessary to take something else, as it was mentioned before, into consideration. This means measuring different values over the circuit and comparing them to the ones obtained in theoretical calculations.

For that purpose, the whole system was implemented into a breadboard, a board that allows connecting components without soldering. This can be really useful, as it is possible to change, remove or add components easily. It also makes it possible to measure the real voltage and current values in a quick way, in order to check them and make adjustments when necessary. In other words, it is the perfect tool for the testing stage of the project.

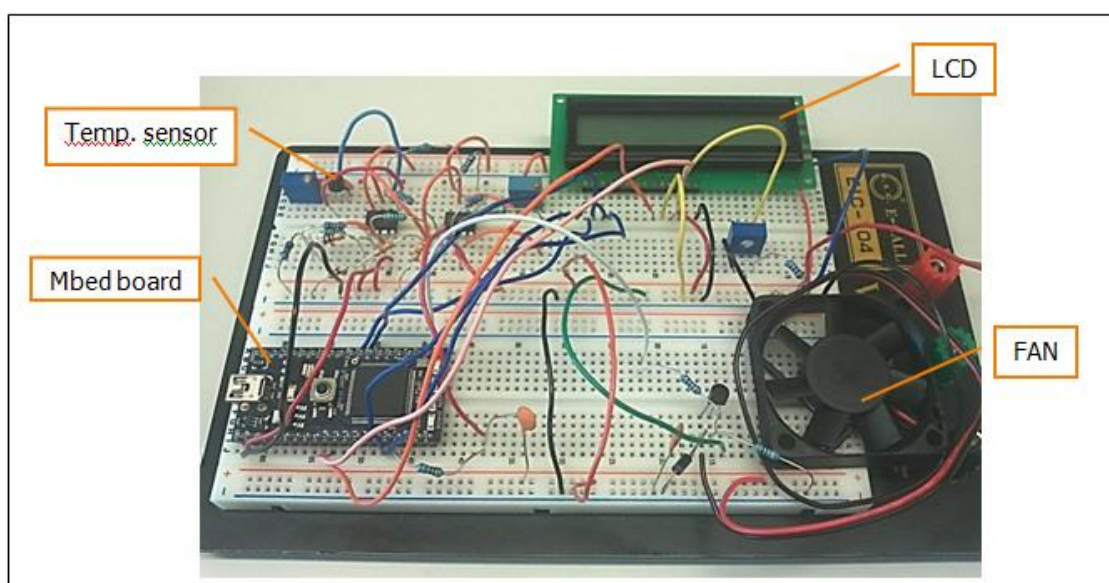


Figure 16. Breadboard with the system implemented on it

The breadboard can be seen in figure 16, once the whole system was implemented on it, including the mbed board, the fan and the LCD.

4.1 Temperature Sensor

As some of the values to be measured may vary depending on temperature, it was required to simulate changes. For this purpose, as the first approach, instead of using the temperature sensor, a voltage generated with a power source was used. This allows to check properly the result obtained at different temperatures in an easy and quickly way. To implement this, the power source available at the laboratory was connected in the points the sensor is supposed to be, so different voltage values can be applied, simulating different temperatures.

To measure the values of the voltages, the multimeter provided at the laboratory was used. This way it was confirmed, as expected, that some of the values slightly differ from theoretical calculations.

It was found that the voltage generated with the regulator is somewhat smaller due to the values of the components chosen and their tolerance. Instead of 1.27 V according to theory, a value of 1.17 V was measured. This affects the value of the resistors, as seen in equations (8) and (11), having to adjust them to a new value.

Table3. Values obtained for the resistors according to the parameters

	SETTINGS						
	T ^a	VCTRL	ADC Max	A	β (ADC Out)	R1 (offset)	R4 (slope)
1st STEP	0	1,17	3,3	0,00273	-	428,57	-
2nd STEP	100	1,17	3,3	0,00373	3,3	-	3300,00

It was also discovered that the operational amplifiers chosen for the project introduce some error, when checking the values in their input pins and the voltage obtained in their output pin. This causes small variations in the output signal, which means errors that affect the accuracy of the system.

It was also needed to include an RC filter, to remove the noise present on the signal before it goes to ADC input.

Then, once the system was tested this way, the sensor was included, in order to test the whole system itself. To simulate temperature changes in the sensor, a heater was used, at the same time temperature was measured with a thermocouple connected to a multimeter, so that it was possible to compare the temperature value obtained by the system and the temperature value measured by the thermocouple, showed in the multimeter. How the connection was implemented is shown in figure 17. The heating tool is described in detail in the corresponding section.

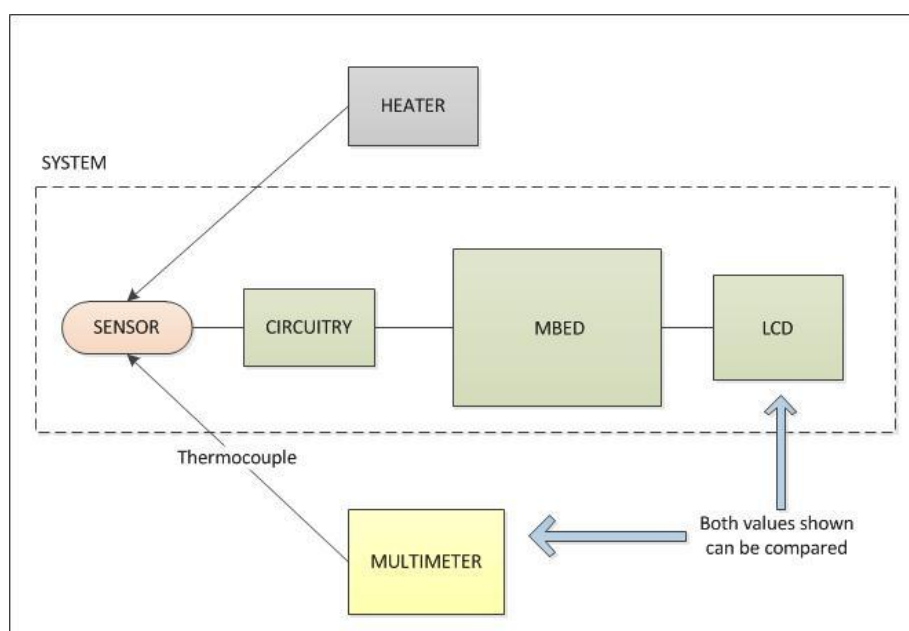


Figure 17. Block diagram with the connection of the tools used for testing

One of the challenges that arose while testing was that the temperature sensor chosen for the project, LM335, reacts slowly to temperature changes, as can be seen in figure 18. It needs some minutes to reach the final temperature value, whereas a thermocouple is very sensitive, the value showed in the multimeter was constantly changing. So when using the heating tool that has a similar functioning than a hairdryer, which means that simply with redirecting the air flow slightly, there were small but noticeable changes in the measurement results.

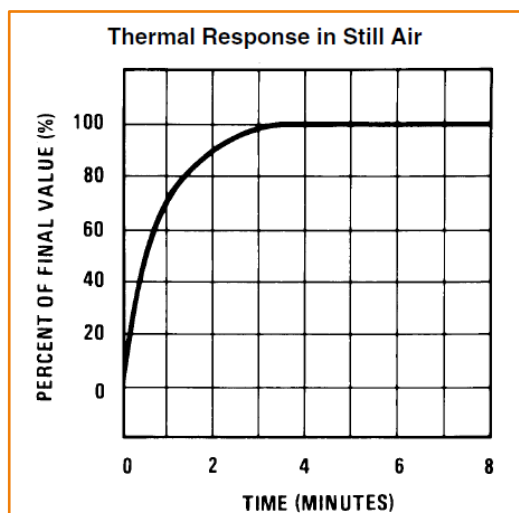


Figure 18. LM335 temperature sensor response in time. Modified from Texas Instruments [7]

As a result, it was only possible to check that both temperature values, the one obtained with the system and the one measured by the thermocouple roughly match, with an average error of 1°C , but there were some sudden changes up to four degrees, especially when reaching the highest temperatures of the system range.

4.2 DC Motor

To test this part, it was completely necessary to use the microcontroller together with the hardware part concerning the DC motor, so that it was possible to apply different PWM signals, testing the behavior when changing the values of the frequency and duty cycle, to check that the system was working as expected, according to theory mentioned in the previous section.

To check the PWM signal, as well as the signal generated with the MOSFET to control the DC motor, the oscilloscope was used. Below can be seen the result obtained for a duty cycle of 30% and 80%.

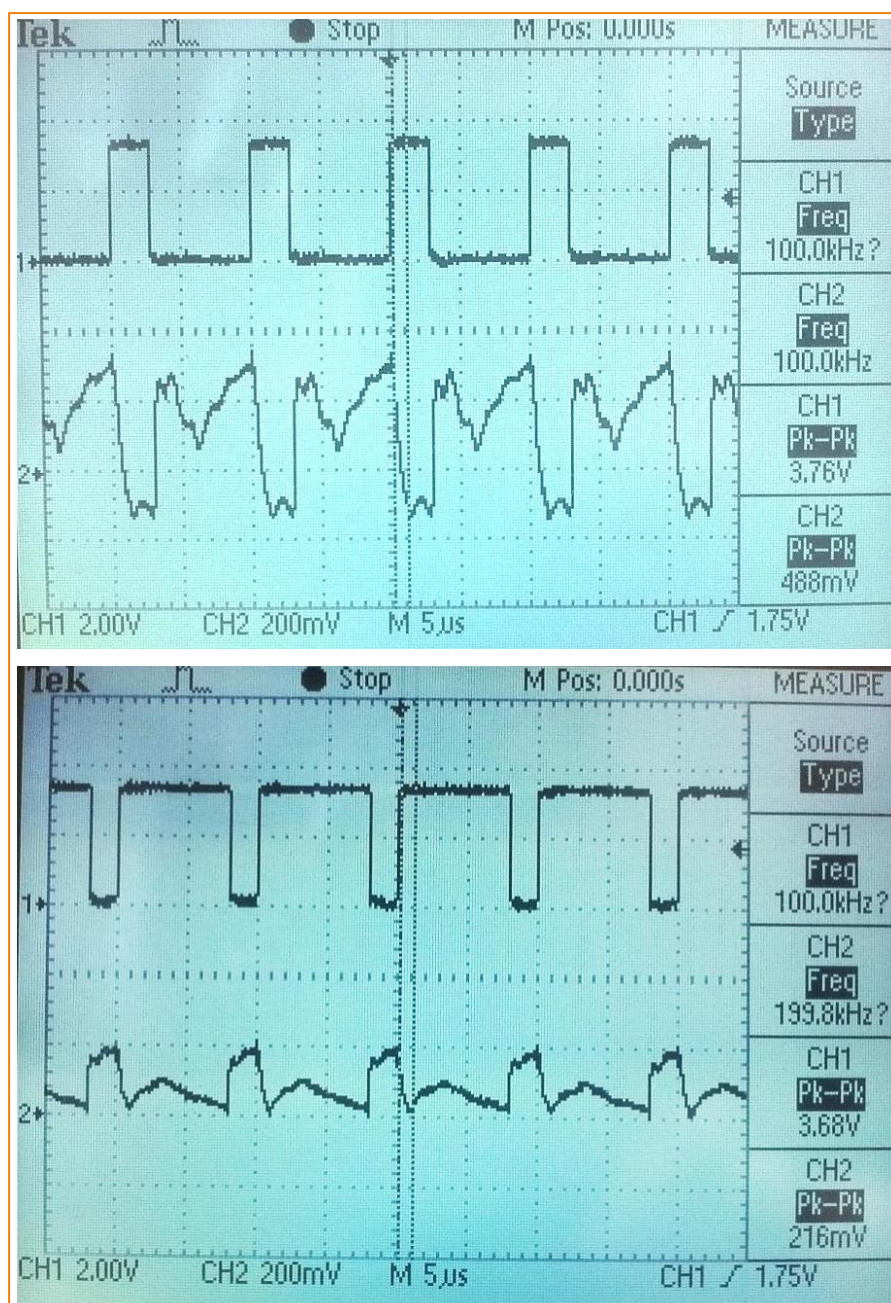


Figure 19. Oscilloscope screenshots

In the figure above, there are two screenshots of the oscilloscope. The PWM signal is in channel 1, and the voltage between drain and source (V_{DS}) of the MOSFET is shown in channel 2. As can be seen there is a significant change in V_{DS} when the width of the PWM also changes.

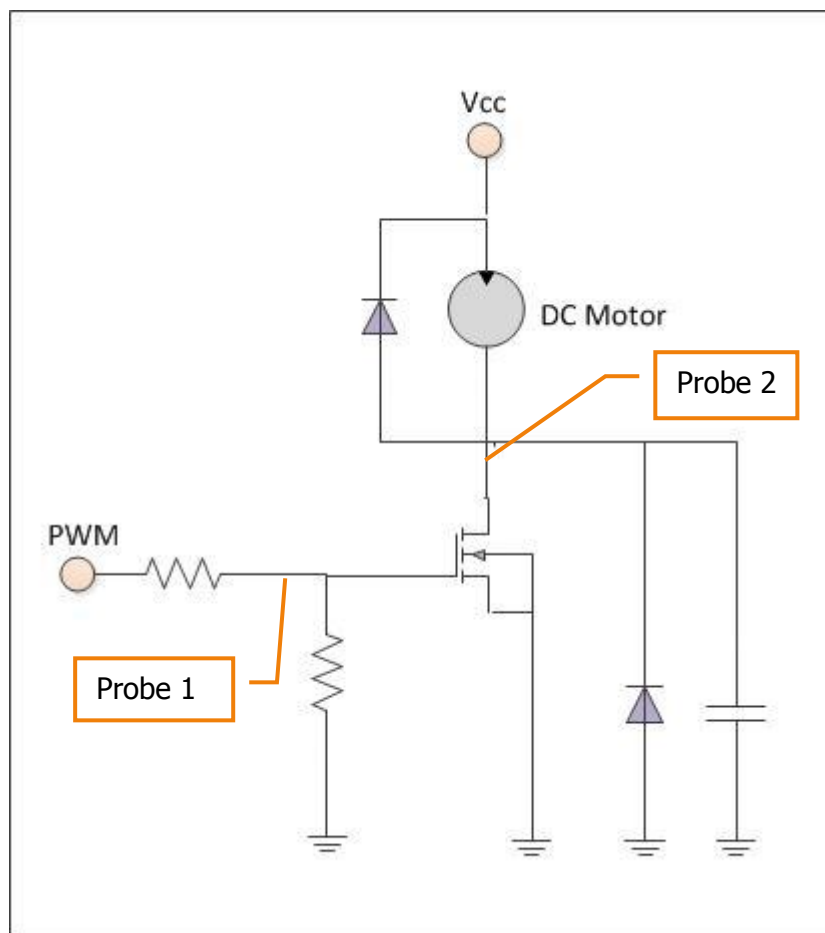


Figure 20. Diagram of connecting points of the oscilloscope probes

In the figure above, there is a diagram explaining where the probes of the oscilloscope where connected to obtain the signals shown in figure 19.

4.3 LCD Display

The LCD display may look it is not as important as the rest of the system, but it also requires to be checked. Once the contrast was properly adjusted using a potentiometer, so that information shown in the display can be read, a testing code was written for the microcontroller to check the LCD pins were properly connected.

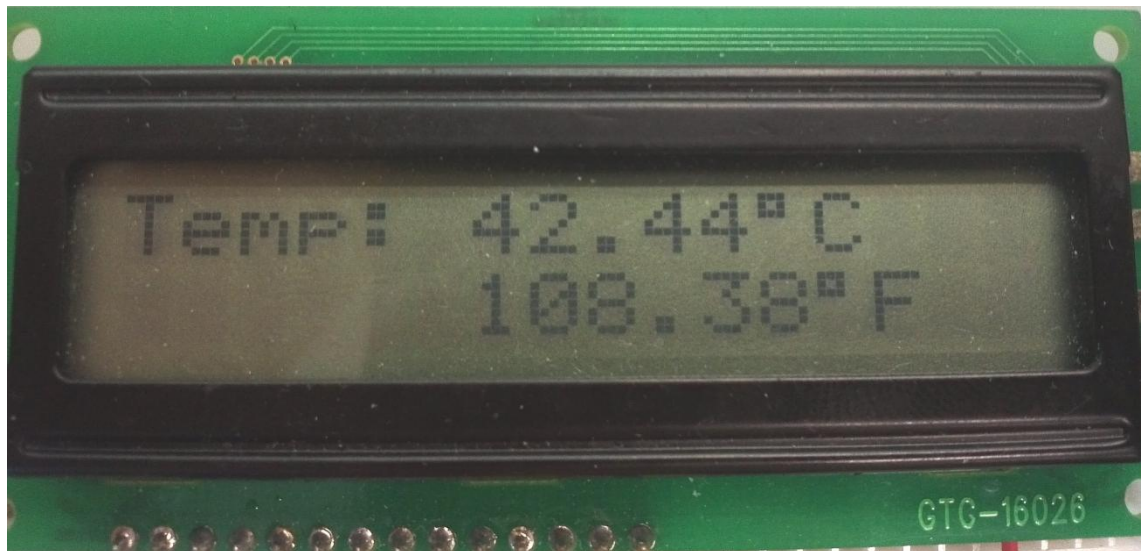


Figure 21. LCD Display testing

As shown in the figure above, in the upper row temperature is displayed in Celsius degrees and in the lower row can be seen in Fahrenheit degrees.

5 Board Designing

Having the system implemented in a breadboard it is useful to check and make changes, but once everything seems to be right, prototype should be implemented in a proper board. After testing the system, it was time to design the printed circuit board (PCB), a board made of glass fiber, with printed copper lines connecting the components instead of the wires used in the breadboard. This kind of board is much more reliable and resistant, but as the design is printed into it, it is almost impossible to make changes on it, so it has to be made once the whole system has been properly checked, making sure everything is alright.

Nowadays, as electronic circuits design has expanded exponentially, there are several different computer programs to design this type of boards. There are even some online applications that require no installation.

The program available at the laboratory, PADS, is software created by Mentor Graphics® that allows creating all the files needed to develop the prototype on a PCB, which means schematic, layout and routing.

Next, there is a brief summary of the different tools provided with this software, and how they were used through the several steps taken when developing the PCB.

5.1 Schematic

The first one, schematic, is the file that contains all the characteristics of the components, such as number of pins or encapsulation as well as how components are connected between them in a clear way, letting other people study the design in the future easily. A screenshot of PADS program showing this file can be seen in figure 22.

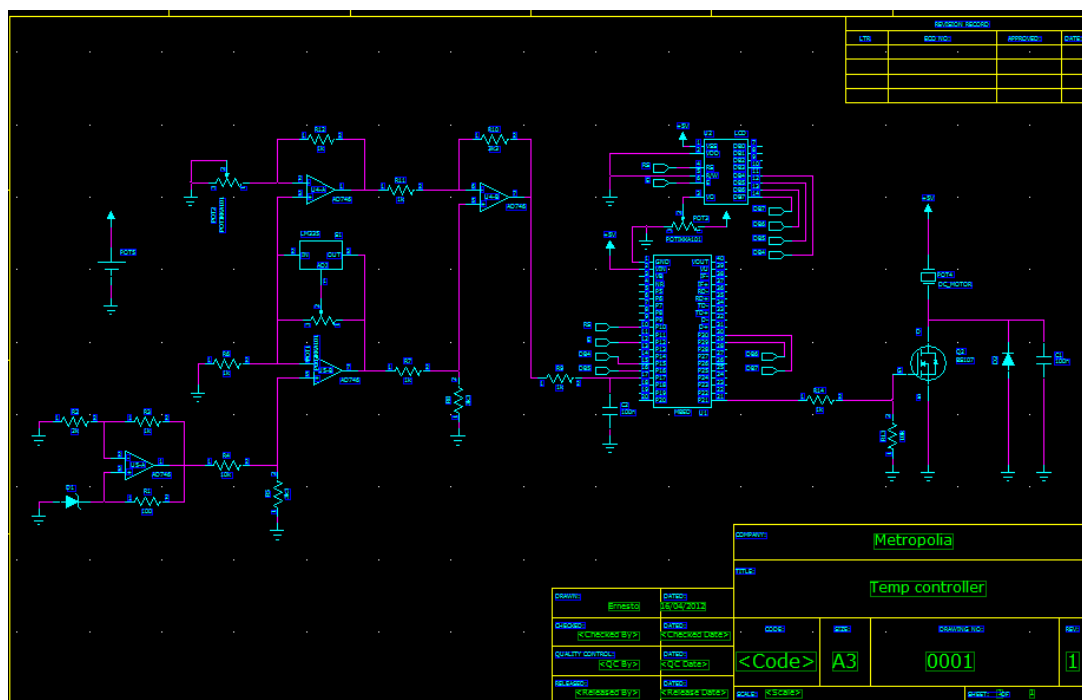


Figure 22. Screenshot from PADS Logic showing the schematic file [14]

One of the challenges found in this part was that many of the components needed for the project were not included into the libraries, which only have the most common components. So it was required to create a new library with the missing components, as for example the mbed board or the LCD. For this purpose PADS provides a part editor that allows the user to create new components, choosing their characteristics.

5.2 Layout

Once the schematic file is done, is time to make the layout. This file contains how components are distributed through the board and allows choosing the characteristics of the board itself, as for example, size or number of layers.

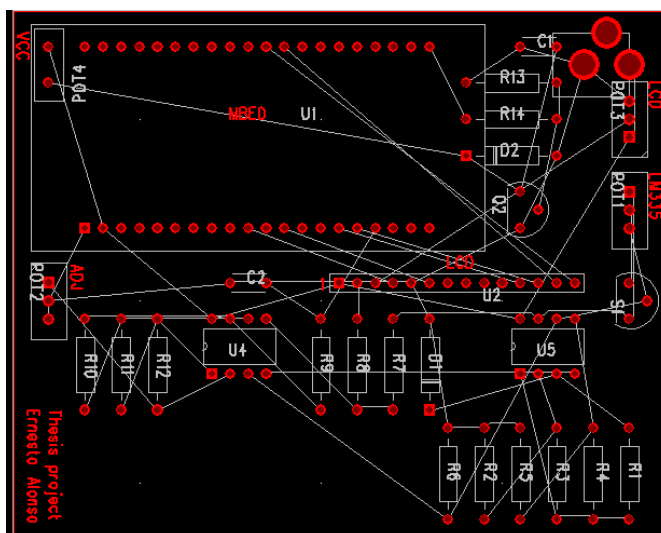


Figure 23. Screenshot from PADS Layout showing the components once rearranged on the board [15]

It takes time to find a proper location for the components, as they should be close to the components to which they are connected according to the schematic, as well as there are several rules to be followed.

Next step is connecting the components placed in the board, as the schematics file says, drawing traces between the pins of the components. This task can be a bit complex if done by hand. PADS provides a tool called autorouter, which simplifies this step, drawing the traces automatically, leaving for the user the task of checking and redrawing certain traces, if necessary, by hand.

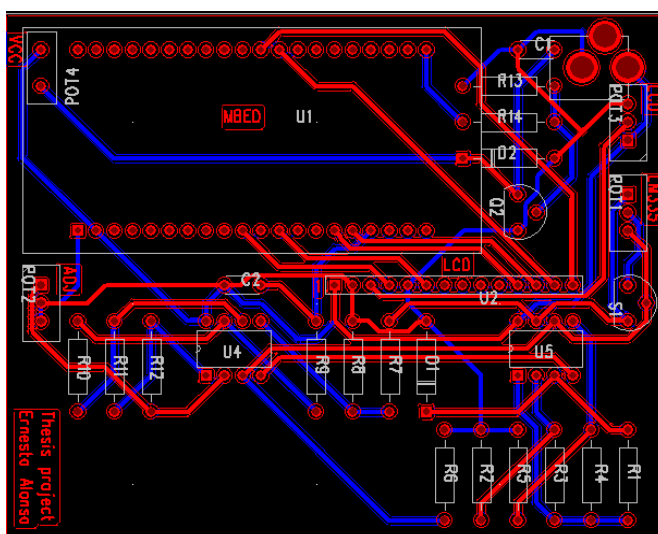


Figure 24. Screenshot from PADS Layout showing the board once routed [15]

Sometimes it is necessary to unroute, in order to relocate some components, so that traces can fit better. The objective is to make traces as straight and short as possible whenever they follow certain design rules. In this project, due to the complexity of the design, it was necessary to employ two layers, as it was entirely impossible to fit it in only one. This fact can be appreciated in figure 24 as traces are drawn in two different colours, red for traces on top layer and blue for the ones placed on the bottom layer.

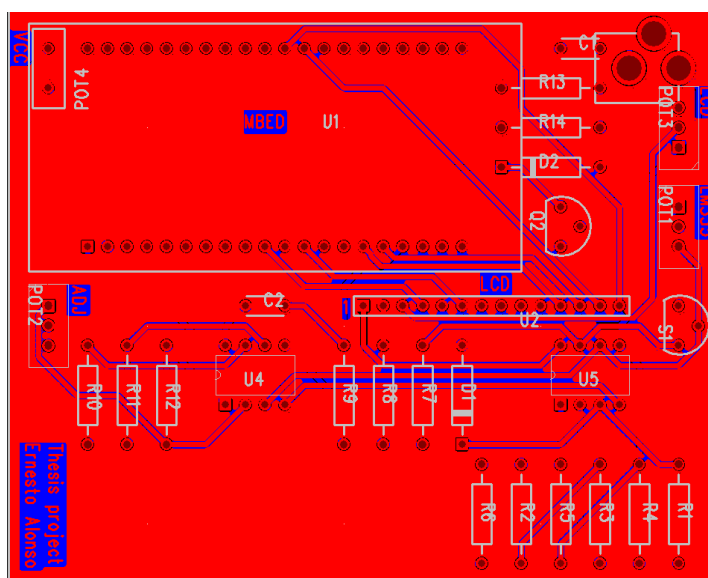


Figure 25. Screenshot from PADS Layout showing the board after creating the ground plane.
[15]

Once the design has been routed properly, those parts of the board that are empty are filled, creating a ground plane. This is shown in the figure above.

5.3 CAM and Printing

The last step is creating the files needed by the exposure unit and the drilling machine, so that this will permit to make the board according to the information contained in the files mentioned before.

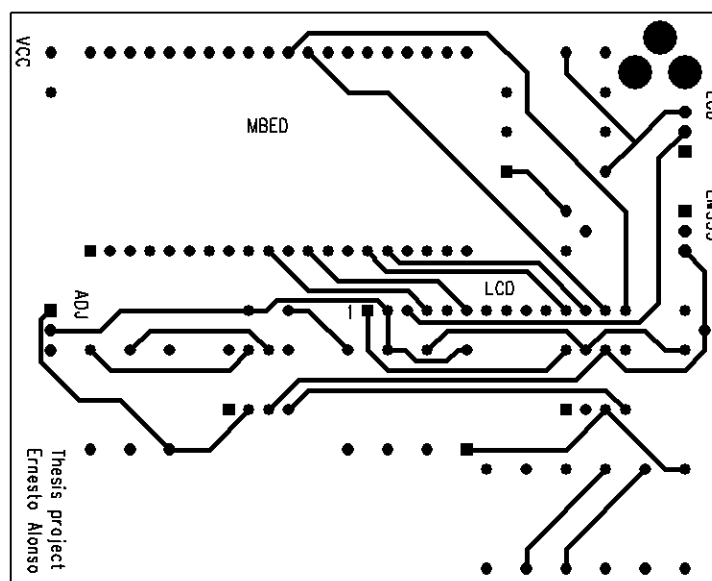


Figure 26. Screenshot from PADS Layout. Preview of how the design will look on the board viewed from top layer. [15]

There is one file for each layer, two in this case, containing the information of the position of the traces and the pads, points where components are inserted, except on the case of the file for the drilling machine as the holes have to match in both layers.

It is also possible, and highly recommended, to print the design, to check and verify the dimensions of the board and location of the components. In figure 26, we can see a preview that was printed in fact, letting check that the dimensions of the board will be 9 x 7.25 cm.

6 Materials

After explaining the whole process followed to develop and test the system, is time to introduce the most important tools that took part in it. All of them were provided at the laboratory of Metropolia campus in Albertinkatu.

As explained in the testing section, to simulate temperature changes it was necessary to use a heater. For this project the hot-air station Hakko 850 was provided at the laboratory. It can be seen in figure 27.



Figure 27. Hot-air station Hakko 850. Reprinted from Hakko USA [16]

As well, to measure the temperature at the same time, in order to compare the value obtained with the system to see the difference between them, a thermocouple connected to a multimeter was used. Due to the fact that a special connector is needed for the thermocouple, it was necessary to use multimeter mastech my64, as not all multimeters available at the laboratory have this kind of connector.

A thermocouple is a device with an alloy of two metals that produces a voltage between two terminals depending on temperature.

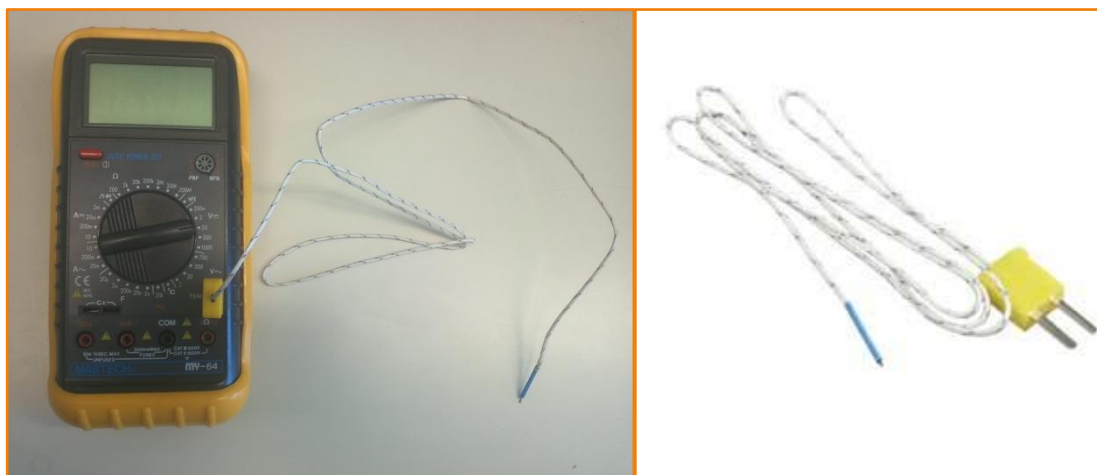


Figure 28. Multimeter with thermocouple connected(left). Detail of the thermocouple connector (right)

Both items, multimeter and thermocouple, are shown connected in figure 28 on the left. On the right, the connector for the thermocouple can be appreciated.

7 Discussion

7.1 Results and Issues

The main purpose of a thesis project is to face the different issues that are involved through all the steps when designing and developing an electronic system. This implies making decisions, scheduling the different tasks to be done, analyzing the results that are obtained, and as a result, finding a solution.

In this section, the objective is to summarize the different challenges that showed up while developing and designing the project, and how they were solved to obtain a successful solution, according to the requirements.

One of the objectives pursued in this project was to create an accurate rotation control system, depending on the temperature measured. So the most important challenge found was to design the electronics surrounding the temperature without introducing bigger errors than the ones that the sensor itself introduces. According to the datasheet, that can be seen in appendix 3, the typical error given by the temperature sensor once calibrated is 1° C.

This was roughly achieved, due to several error sources. One of them is the regulator designed, as it was explained in section 3. Due to the values, and tolerances, of the resistors the voltage obtained is slightly smaller than expected. The operational amplifiers chosen for the project are the other main focus of error. They introduce certain errors that cause small variations in the voltage values, making them differ from the theoretical ones.

As a result, the fan has a higher rotation speed as temperature rises, or the other way round, the fan slows down when temperature decreases. Changes in the rotation speed of the fan are only noticeable when significant changes of temperature happen, so the only possible way to check the accuracy of the system is by looking at the values showed in the LCD display.

7.2 Future Development

Due to the lack of time it was not possible to study in depth the errors that introduce the operational amplifiers and how this affects the accuracy of the system. It would be interesting to study the possibility of changing them for instrumentation amplifiers, and how much this improves the results obtained, compared to the ones showed in this report.

Another change that can be introduced, is designing a step-up converter, so that the DC motor is supplied with 12 V, which is the rated voltage according to the datasheet included in appendix 6, increasing the airflow of the fan. It would be necessary to study how this increment affects the energy losses, and as result PWM frequency might be changed.

As an improvement, it would be interesting to create a user interface, allowing choosing in which range of temperature the system should work. In other words, making it possible to choose at which temperature the fan begins to move, and at which one it reaches the maximum speed. The MBED board also includes a serial interface, so it would be possible to develop another external board with a display and some buttons, and communicate both boards through this serial interface mentioned. It would also be necessary to modify the code shown in appendix 1, to include a function to attend the serial communication, as well as modify the existing functions to fit the new characteristic.

8 Conclusion

Going back to the introduction, it was said that the goal was to develop a system controlled by a microcontroller, to see how they simplify and perform a task that at a first sight looks much more complicated.

The objective was achieved, as the system developed in this project controls the speed rotation of the fan as required, as well as the temperature value measured, with a reasonable error, can be seen in the LCD display. In the figure below the system can be seen working at room temperature.

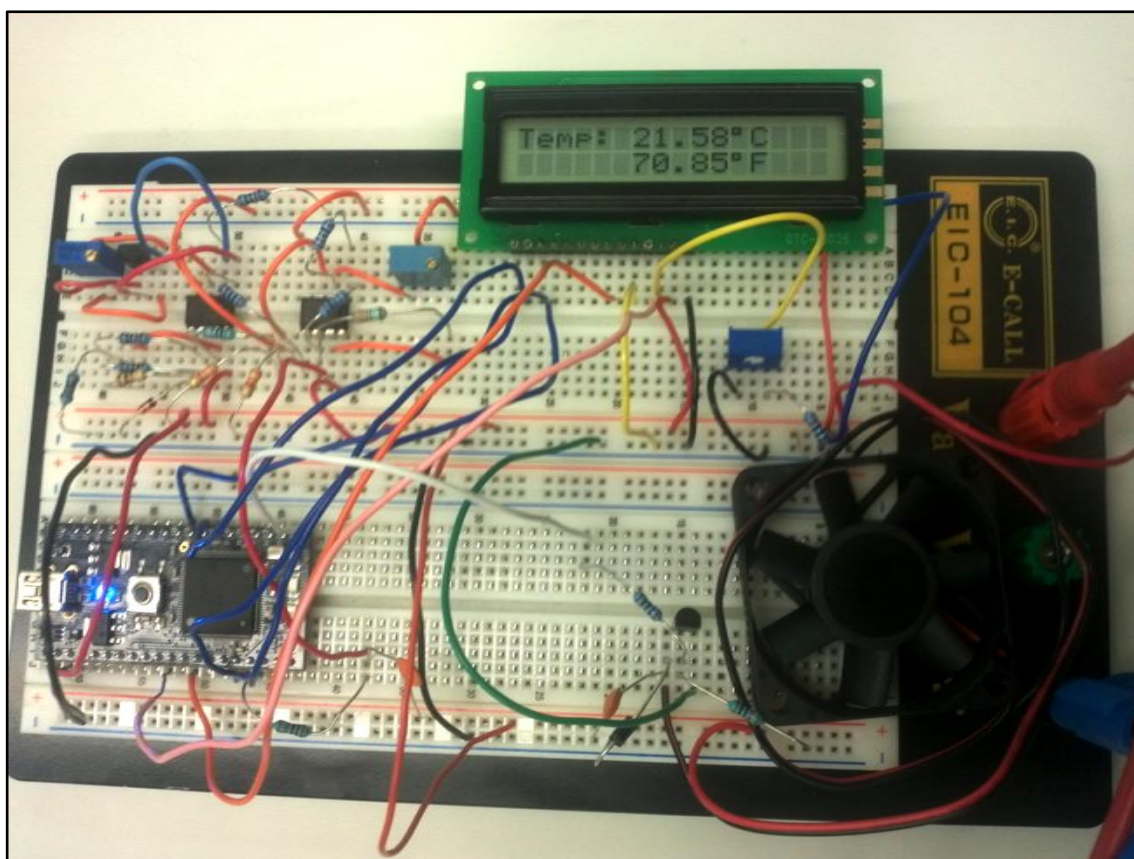


Figure 29. System working at room temperature

Even though the system developed reaches the requirements set on the specifications, it can probably be improved with a further research on the error sources, as one of the goals was to create an accurate system, and although this prototype can be considered as such, it can also reach a higher accuracy level.

References

1. Ernest Braun, Stuart Macdonald. Revolution in Miniature: The History and Impact of Semiconductor Electronics. Cambridge, Press Syndicate of the University of Cambridge; 1978.
2. Wayne Storr. Electronics Tutorial about MOSFETs as Switches [online]; May 2012.
URL: http://www.electronics-tutorials.ws/transistor/tran_7.html
Accessed May 5th, 2012
3. Bonnie Baker. Designing with temperature sensors, part one: sensor types [online]; September 2011.
4. Robert J. Distinti. Intellectual Property Development Associates of Connecticut, Trumbull. Analog to Digital Converter. US patent 5,202,687; 1991.
5. Len Staller. Understanding analog to digital converter specifications [online]; February 2005.
URL: <http://www.eetimes.com/design/embedded/4025078/Understanding-analog-to-digital-converter-specifications>
Accessed May 5th, 2012
6. National Instruments. Pulse Width Modulation (PWM) Using NI-DAQmx and LabVIEW [online]; May 2010.
URL: <http://www.ni.com/white-paper/2991/en>
Accessed May 5th, 2012
7. National Semiconductor Corporation. Datasheet: LM135/LM235/LM335, LM135A/LM235A/LM335A Precision Temperature Sensors; December 17, 2008.
URL: <http://www.ti.com/lit/ds/symlink/lm335.pdf>
Accessed May 5th, 2012

8. Ward Brown. Brushless DC motor control made easy [online]. Microchip; 2002.
URL: <http://www.ti.com/lit/ds/symlink/lm335.pdf>
Accessed May 5th, 2012
9. Simon Ford, Chris Styles. Mbed overview [online]
URL: <http://mbed.org/handbook/mbed-NXP-LPC1768>
Accessed May 5th, 2012
10. Simon Ford, Chris Styles. Mbed about [online]
URL: <http://mbed.org/handbook/About>
Accessed May 5th, 2012
11. Simon Ford, Chris Styles. Mbed compiler [online]
<http://mbed.org/handbook/mbed-Compiler>
Accessed May 5th, 2012
12. Keil uVision [computer program]. Version 4. Germany: ARM; September 2009
13. Simon Ford, Chris Styles. Mbed SDK [online]
<http://mbed.org/handbook/mbed-SDK>
Accessed May 5th, 2012
14. PADS Logic [computer program]. Version 9.2. Wilsonville, OR: Mentor Graphics; 2010
15. PADS Layout [computer program]. Version 9.2. Wilsonville, OR: Mentor Graphics; 2010
16. Hakko USA. Hakko 850 Modification for Nitrogen Supply [online]; May 2010
URL: <http://www.hakkousa.com/kb/KnowledgebaseArticle10421.aspx>
Accessed May 5th, 2012

Appendices

Appendix 1. Temperature Sensor LM335 Datasheet

Temperature Accuracy (Note 1)

LM335, LM335A

Parameter	Conditions	LM335A			LM335			Units
		Min	Typ	Max	Min	Typ	Max	
Operating Output Voltage	$T_C = 25^\circ\text{C}$, $I_R = 1\text{ mA}$	2.95	2.98	3.01	2.92	2.98	3.04	V
Uncalibrated Temperature Error	$T_C = 25^\circ\text{C}$, $I_R = 1\text{ mA}$		1	3		2	6	$^\circ\text{C}$
Uncalibrated Temperature Error	$T_{\text{MIN}} \leq T_C \leq T_{\text{MAX}}$, $I_R = 1\text{ mA}$		2	5		4	9	$^\circ\text{C}$
Temperature Error with 25°C Calibration	$T_{\text{MIN}} \leq T_C \leq T_{\text{MAX}}$, $I_R = 1\text{ mA}$		0.5	1		1	2	$^\circ\text{C}$
Calibrated Error at Extended Temperatures	$T_C = T_{\text{MAX}}$ (Intermittent)		2			2		$^\circ\text{C}$
Non-Linearity	$I_R = 1\text{ mA}$		0.3	1.5		0.3	1.5	$^\circ\text{C}$

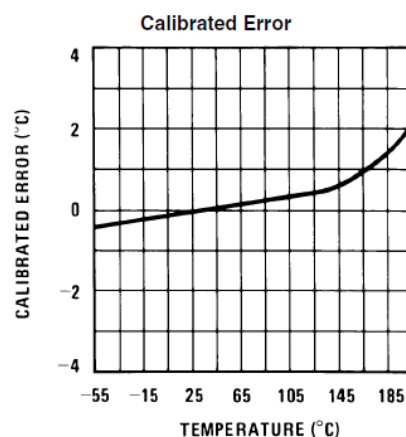
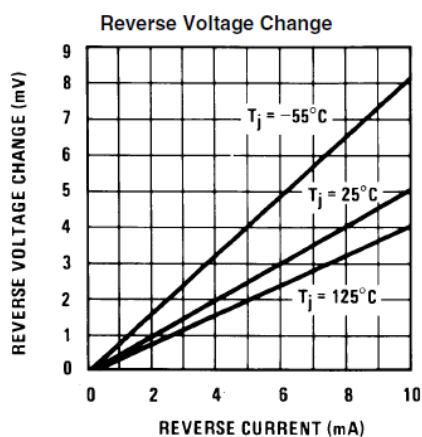
Electrical Characteristics (Note 1)

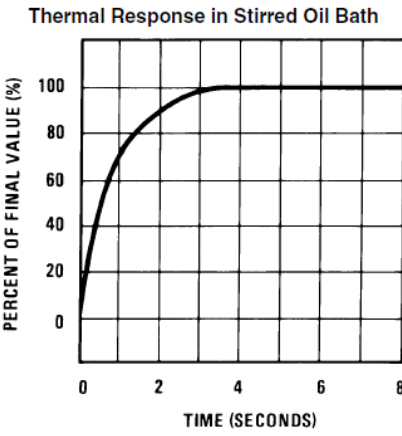
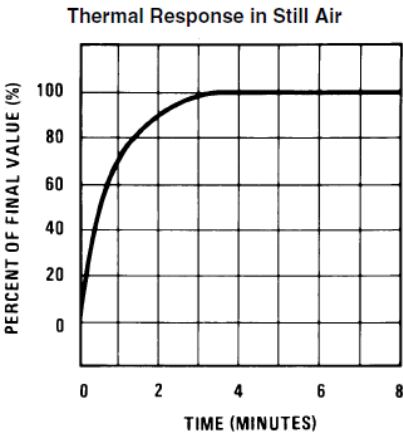
Parameter	Conditions	LM135/LM235 LM135A/LM235A			LM335 LM335A			Units
		Min	Typ	Max	Min	Typ	Max	
Operating Output Voltage	$400\text{ }\mu\text{A} \leq I_R \leq 5\text{ mA}$		2.5	10		3	14	mV
Change with Current	At Constant Temperature							
Dynamic Impedance	$I_R = 1\text{ mA}$		0.5			0.6		Ω
Output Voltage Temperature Coefficient			+10			+10		$\text{mV}/^\circ\text{C}$
Time Constant	Still Air		80			80		sec
	100 ft/Min Air		10			10		sec
	Stirred Oil		1			1		sec
Time Stability	$T_C = 125^\circ\text{C}$		0.2			0.2		$^\circ\text{C}/\text{hr}$

www.national.com

2

Typical Performance Characteristics





Appendix 2. ADC Characteristics

12. ADC electrical characteristics

Table 18. ADC characteristics (full resolution)*V_{DDA} = 2.7 V to 3.6 V; T_{amb} = -40 °C to +85 °C unless otherwise specified; ADC frequency 13 MHz; 12-bit resolution.*

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V _{IA}	analog input voltage		0	-	V _{DDA}	V
C _{ia}	analog input capacitance		-	-	15	pF
E _D	differential linearity error		[1][2]	-	±1	LSB
E _{L(adj)}	integral non-linearity		[3]	-	±3	LSB
E _O	offset error		[4][5]	-	±2	LSB
E _G	gain error		[6]	-	0.5	%
E _T	absolute error		[7]	-	4	LSB
R _{vsi}	voltage source interface resistance		[8]	-	7.5	kΩ
f _{clk(ADC)}	ADC clock frequency		-	-	13	MHz
f _{o(ADC)}	ADC conversion frequency		[9]	-	200	kHz

[1] The ADC is monotonic, there are no missing codes.

[2] The differential linearity error (E_D) is the difference between the actual step width and the ideal step width. See [Figure 26](#).[3] The integral non-linearity (E_{L(adj)}) is the peak difference between the center of the steps of the actual and the ideal transfer curve after appropriate adjustment of gain and offset errors. See [Figure 26](#).[4] The offset error (E_O) is the absolute difference between the straight line which fits the actual curve and the straight line which fits the ideal curve. See [Figure 26](#).[5] ADCOFFS value (bits 7:4) = 2 in the ADTRM register. See *LPC17xx user manual UM10360*.[6] The gain error (E_G) is the relative difference in percent between the straight line fitting the actual transfer curve after removing offset error, and the straight line which fits the ideal transfer curve. See [Figure 26](#).[7] The absolute error (E_T) is the maximum difference between the center of the steps of the actual transfer curve of the non-calibrated ADC and the ideal transfer curve. See [Figure 26](#).[8] See [Figure 27](#).

[9] The conversion frequency corresponds to the number of samples per second.

Appendix 3. Temperature sensor circuitry

SETTINGS							
	T^a	VCTRL	ADC Max	α	β (ADC Out)	R1 (offset)	R4 (slope)
1st STEP	0	1,25	3,3	0,00273	-	457,88	-
2nd STEP	100	1,25	3,3	0,00373	3,3	-	3300,00

T^a	LM335 (V)	V+	V-	Vo (V)	ADC Code
0	2,73	3,98	1,25	0,00	0
5	2,78	4,03	1,25	0,17	13
10	2,83	4,08	1,25	0,33	26
15	2,88	4,13	1,25	0,50	38
20	2,93	4,18	1,25	0,66	51
25	2,98	4,23	1,25	0,83	64
30	3,03	4,28	1,25	0,99	77
35	3,08	4,33	1,25	1,16	90
40	3,13	4,38	1,25	1,32	102
45	3,18	4,43	1,25	1,49	115
50	3,23	4,48	1,25	1,65	128
55	3,28	4,53	1,25	1,82	141
60	3,33	4,58	1,25	1,98	154
65	3,38	4,63	1,25	2,15	166
70	3,43	4,68	1,25	2,31	179
75	3,48	4,73	1,25	2,48	192
80	3,53	4,78	1,25	2,64	205
85	3,58	4,83	1,25	2,80	218
90	3,63	4,88	1,25	2,97	230
95	3,68	4,93	1,25	3,13	243
100	3,73	4,98	1,25	3,30	256

Appendix 4. Source Code

```

#include "mbed.h"
#include "TextLCD.h"

AnalogIn temperature(p17); //Variable for analog input (ADC). Pin
17
PwmOut motor(p21); //Variable for PWM output. Pin 21
TextLCD lcd(p10, p12, p15, p16, p29, p30, TextLCD::LCD16x2);
//Variable for LCD. Pins for rs, e, d4-d7

//Function that converts the voltage acquired to temperature
float ADC_to_temp ()
{
    int i;
    float media = 0.0;
    float temp;

    //30 samples are taken, in order to have more accurate results
    for(i=0; i<30; i++)
    {
        media = media + temperature.read();
    }
    media = media / 30;
    temp = media * 100;
    return (temp);
}

//Function that generates a PWM signal with a duty cycle depending
on temperature
void PWM_generator(float temp)
{
    motor.period_us(10); //freq = 100 kHz
    motor = temp / 100;
}

void Print_LCD(float temperature, float fahrenheit)
{
    lcd.printf("Temp: %.2f°C \n", temperature, 223);
    lcd.locate(6,1); //Second row, 7th column
    lcd.printf("%.2f°F \n", fahrenheit, 223);
}

```



```
int main()
{
float temperature;
float fahrenheit;
int i;

while(1)
{
temperature = ADC_to_temp();

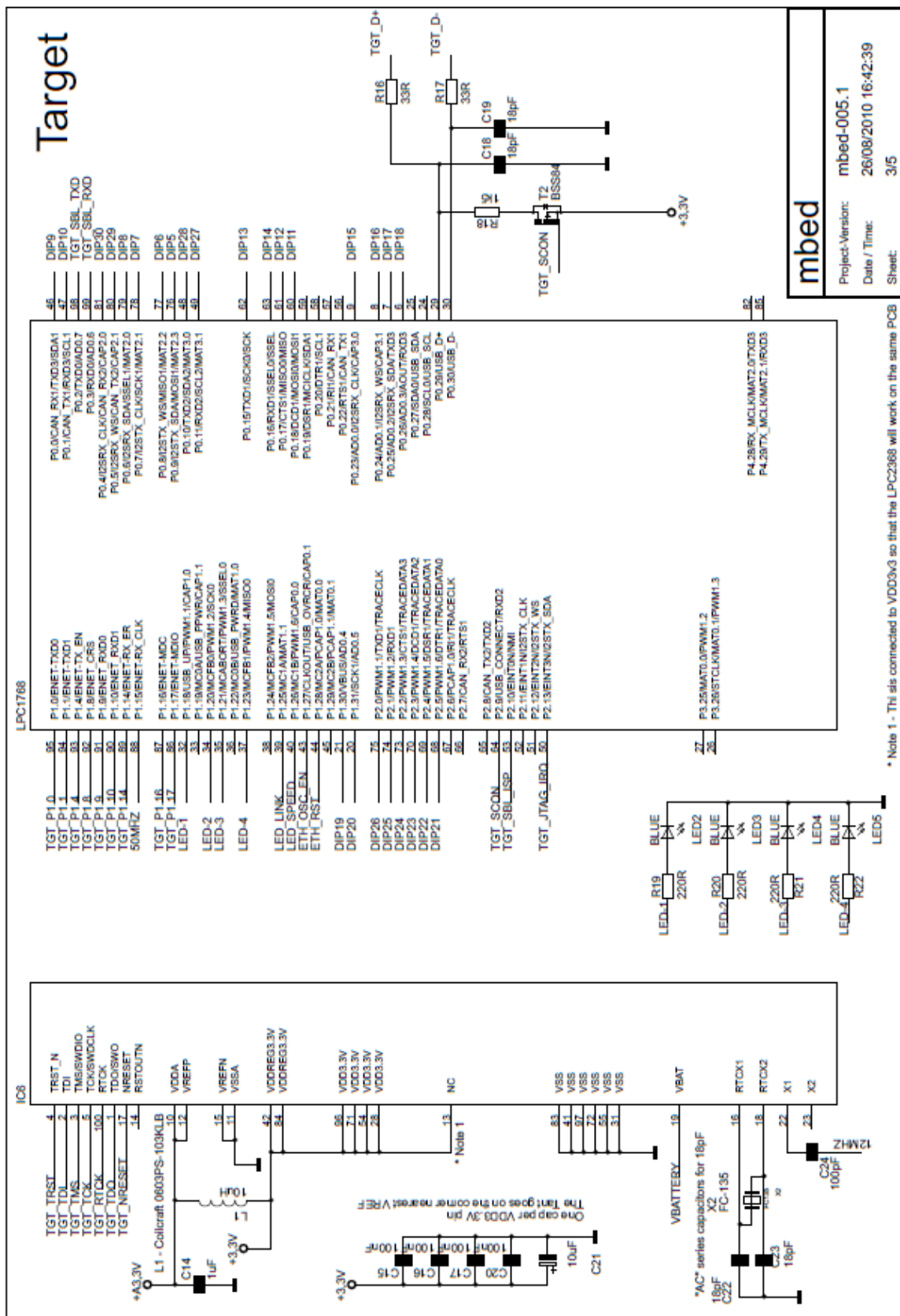
PWM_generator(temperature);

fahrenheit = temperature * 1.8 + 32;

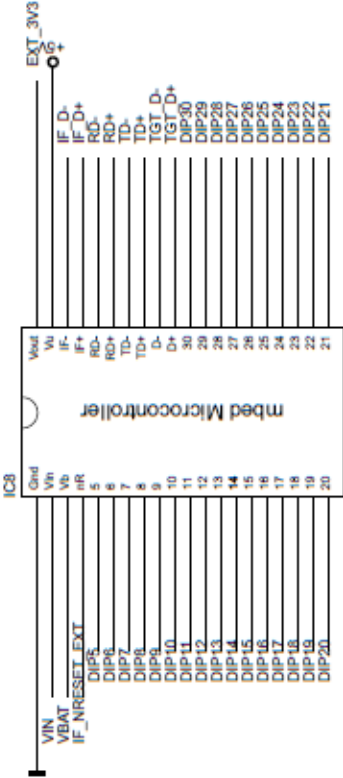
Print_LCD(temperature, fahrenheit);
wait(5);

}
}
```

Appendix 5. Mbed board schematic



DIP Pinout



mbed	
Project Version:	mbed-005.1
Date / Time:	26/08/2010 16:42:39
Sheet:	5/5

Appendix 6. DC Motor (fan) datasheet

SUNON**SPECIFICATIONS****MODEL : KDE1204PKVX****P/N : MS.AF.GN**

1-1. Rated Voltage	: 12 VDC
1-2. Operating Voltage Range	: 4.5~13.8 VDC
1-3. Starting Voltage	: 4.5 VDC (25 deg. C POWER ON/OFF)
1-4. Rated Speed	: 8200 RPM \pm 15%
1-5. Air Delivery	: 10.8 CFM
1-6. Static Pressure	: 0.27 Inch-H ₂ O
1-7. Rated Current	: 0.12 AMP
1-8. Rated Power	: 1.4 WATTS
1-9. Noise Level	: 27.5 dB(A)
1-10. Direction of Rotation	: Counter-clockwise viewed from front of fan blade
1-11. Operating Temperature	: -10 to +70 deg. C
1-12. Storage Temperature	: -40 to +70 deg. C
1-13. Bearing System	: Vapo bearing system
1-14. Weight	: 31 g
1-15. Safety	: UL/CUR/TUV/CE Approvals
1-16. Vibration	: Vibration of acceleration 1.5G and frequency 5~50~5Hz is applied in all 3 directions(X,Y,Z), in cycles of 1 minute each, for a total vibration time of 30 minutes.
1-17. Locked Rotor Protection	: Automatic Restart Capability Note: In a situation where the fan is locked by an external force while the electricity is on, an increase in coil temperature will be prevented by temporarily turning off the electrical power to the motor. The fan will automatically restart when the locked rotor condition is released.

建準電機工業股份有限公司
SUNONWEALTH ELECTRIC
MACHINE INDUSTRY CO., LTD.

SPEC.NO	D04040170G-00	REVISION D
ISSUE DATE	05. 03. 2005	E D I T I O N
P A G E	3 OF 11	

